UK UNCLASSIFIED

SANDERLING

FINAL REPORT : Parts A and B

A Research Study into KBS for Command and Control in Naval and TMD Applications

19980309 269

UK UNCLASSIFIED

U3541

Accession Number: 3541

Publication Date: Apr 01, 1990

Title: SANDERLING Final Report: Parts A and B, Research Study into KBS for Command and Control in Naval and TMD Applications

Personal Author: Various from Logica, British Aerospace & Cambridge Consul.

Corporate Author Or Publisher: Admiralty Research Establishment (ARE), Portsdown, UK Report Number: 706.70040

Comments on Document: Final Report: Part A The Study Method, Part B Definition and Technology Analysis

Descriptors, Keywords: SANDERLING C2 KBS TDS TDP TMD SDI Research Navy Program Goal Evaluation Technology Analysis

Pages: 00112

Cataloged Date: Jun 03, 1992

Contract Number: SLS21C/19

Document Type: HC

Number of Copies In Library: 000001

Record ID: 24029

The SANDERLING Final Report comprises the following three volumes :

- The **EXECUTIVE SUMMARY** provides an overview and summary of the study, including its conclusions and key findings, but not including specific detail on suggested projects;

- **PARTS A & B** cover the method and direction of the study, and include details of the technology analysis as well as the initial thniking behind the projects;

- **PART C** of the Final Report defines the recommended research programme in some detail. It describes suggested projects (including form, content, cost and resources), overall programme structure and recommendations on how to proceed.

SANDERLING

Final Issue

April 1990

SANDERLING Final Report :

PART A The Study Method

Project Title :	SANDERLING
Project Number :	706.70040
Client :	ARE, Portsdown
Contract Number :	SLS21C/19
Authors :	Various from Logica, British Aerospace and Cambridge Consultants Ltd.
Status :	Final

Signed : S. Cogbill

Authorised : P G Jenkins

PART A : The Study Method

TABLE OF CONTENTS

List of Abbreviations

The following is a list of the most frequently used acronyms and short forms in the SANDERLING final report.

ADP	-	Automatic Data Processing
AEA	-	Atomic Energy Authority
AGARD	-	Advisory Group for Aerospace Research and Development (NATO)
AI	-	Artificial Intelligence
AOT3	-	A Division of ARE specialising in trials
ARE	-	Admiralty Research Establishment (Portsdown)
AAW	-	Anti-Air Warfare
ASW	-	Anti-Submarine Warfare
ATMS	-	Assumption-based Truth Maintenance System
AXT	-	Division of ARE, divided into AXT3, AXT5 etc.
BAe (SRC)	-	British Aerospace (Sowerby Research Centre)
BMC^2	-	Battle Management Command and Control
BMC^2D	-	Battle Management Command and Control Demonstrator
BMC^3	-	Battle Management Command, Control and Communications
C^2	-	Command and Control
CADDIE	-	Control and Direction of Distributed Intelligent Entities
CAGE	-	DAI Software
CCIS	-	Command and Control Information System
CCL	-	Cambridge Consultants Limited
CCRP	-	Command and Control Research Programme
CLG	-	Command Language Grammar
CONUS	-	Continental United States of America
DAI	-	Distributed Artificial Intelligence
DFM	-	Data Fusion Module (of the TDS)
DBMS	-	Database Management System
ECM	-	Electronic Counter-Measures
EnFun	-	Enhanced Functionality (in reference to the TDS)
ESPRIT	-	European Special Programme for Research in Information Technology
EXPRO	-	Experimental Programme Research Objectives
GA	-	Genetic Algorithms
GOMS	-	Goals Operators Methods Selection
HCI	-	Human Computer Interaction
HMD	-	Helmet Mounted Display
ICAI	-	Intelligent Computer Aided Instruction
IED	-	Information Engineering Directorate
IPP	-	Impact Point Prediction
ITT	-	Invitation to Tender
JTMS	-	Justification-based Truth Maintenance System
KADS	-	Knowledge Acquisiton and Documentation System (ESPRIT I Project)
KBMS	-	Knowledge Based Management Systems
KBS	-	Knowledge Based Systems
KRL	-	Knowledge Representation Language
MACE	-	DAI Software
MOD (PE)	-	Ministry of Defence (Procurement Executive)
MMI	-	Man-Machine Interface

MUSE	-	A tool for building real-time KBS
NP	-	Non-Polynomial
OpDep	-	Operational Dependency (in reference to the TDS)
PWO	-	Principal Weapons Officer
RAE	-	Royal Aerospace Establishment
RAP	-	Resource Allocation Prototype
RFP	-	Request for Proposal
RIPR	-	Research Initiative in Pattern Recognition
RMS	-	Reason Maintenance Systems
RRASL	-	Reactive Resource Allocation at Single Ship Level
RSRE	-	Royal Signals and Radar Establishment
RV	-	Re-entry Vehicle
SADM	-	Structured Analysis and Design Methods
SAP	-	Situation Assessment Prototype
SDF	-	Shore Development Facility (at ARE)
SDI	-	Strategic Defence Initiative
SDIO	-	Strategic Defence Initiative Office (Based in Washington DC)
SDIPO	-	Strategic Defence Initiative Participation Office (Based in London)
SDS	-	Strategic Defence Systems
SNeBR	-	Semantic Network Processing System with Belief Revision
SP	-	Sanderling Project
SSL	-	Software Sciences Limited
TAG	-	Task Action Grammar
TAKD	-	Task Analysis for Knowledge Descriptions
TDP	-	Technology Demonstrator Programme
TDS	-	Technology Demonstrator System
TEWA	-	Target Evaluation and Weapon Assignment
TMD	-	Theatre Missile Defence
TMDD	-	Theatre Missile Defence Demonstrator
TMS	-	Truth Maintenance System
UIMS	-	User Interface Management System
UKAS	-	United Kingdom Architecture Study
V & V	-	Validation and Verification
VDM	-	Vienna Development Method
Z	-	A Notation for Specifying and Designing Software.

1. INTRODUCTION

1.1 Background

Recognizing the potential to be gained in command and control (C^2) systems from knowledge-based systems (KBS) and the related field of artificial intelligence (AI) the Admiralty Research Establishment (ARE) set up the Technology Demonstrator Programme (TDP) and the associated Technology Demonstrator System (TDS). This work is addressing many of the practical and immediate problems associated with the application of KBS to data fusion in the context of naval C^2. The TDS project is progressing towards sea trials as an engineered data fusion demonstrator. The parallel activity of the TDP is providing the supporting research. In addition, ARE is undertaking other research via prototypes, theoretical studies and experiments aimed at establishing a wide capability in KBS based C^2.

Many of the functional characteristics and technical problems associated with naval C^2 are similar to those in other large and complex military applications. In particular it was realised that there was significant commonality with the Strategic Defence Initiative (SDI)/Theatre Missile Defence (TMD) scenario. As a result, a joint research approach was agreed between ARE and the SDIO/SDIPO. A major part of thisapproach is a programme of research extending over four years. The programme will link to the current ARE research activities and will consist of a number of research projects to be undertaken by ARE, UK industry and universities.

The first phase is a 6 month study to define the contents of the research programme. The study commenced in October 1989 and is code-named SANDERLING. It has been undertaken by a consortium of three companies, working in close association with ARE and the SDIO. The consortium comprises Logica Cambridge Ltd, BAe (Sowerby Research Centre) and Cambridge Consultants Ltd. Logica are the prime contractors and overall project managers.

1.2 Aim

The aim of the SANDERLING study is to generate a KBS research programme in Naval C^2 and TMD applications.

1.3 Results and Report

The study's final report consists of an Executive Summary and three main parts, as follows:

Part A Study Method

This introduces the aim and requirements of the study, the goals and objectives of the work and the approach and methods used.

Part B   Definition and Technology Analysis

This part shows how a set of technology-based research streams for the programme was generated and explores the research issues and priorities for sub-topics within each technology stream. This analysis is complemented by an applications' perspective and culminates in a set of initial ideas for SANDERLING research projects.

Part C   The Research Programme

This part uses the results and data from the earlier work to define and evaluate a set of research projects and to formulate a recommended research programme.

1.4 Part A Contents

This document is Part A of the SANDERLING Final Report, and describes the study method. It comprises the following sections:

- Section 2 sets out the goals and objectives for the study and the programme.
- Section 3 explains the approach used to carry out the study.
- Annex A1 is a glossary of Artificial Intelligence and Computing terms used in the report.
- Annex A2 is a list of references.

1.5 Acknowledgements

The SANDERLING consortium wish to acknowledge the support given to the project by staff at ARE, the SDIPO, the SDIO and their associated contractors. We are also grateful for the help and advice we have received from many others, including our own external consultants on the Technical Review Panel, and the members of the Electronic and Business Equipment Association, who contributed valuable background material to the study.

Input from all these sources has contributed significantly to the results of the study and to the shape and form of the proposed research projects and programme. We have naturally been particularly concerned to respond to the comments and feedback from reviews carried out by the SDIO and ARE and to work with them on the design of the projects and programme. However, as always with such tasks, only a co-ordinated proportion of such input can finally be included.

2. GOALS AND OBJECTIVES

2.1 Study Terms of Reference

This section summarises the scope and assumptions of the study.[1]

The SDIO and ARE are seeking from the study a set of recommended research projects that:

- have a combination of near-term and long-term objectives;
- indicate goals and the manner in which they are to be investigated;
- have an identified cost-effectiveness or risk/reward ratio;
- present options to a value in the region of £15M from which a programme can be constructed;
- represent an interlocking set of topics with a clear strategy to guide choice;
- consider available UK expertise.

The resulting Command and Control Research Programme (CCRP) is planned to cover an elapsed time of three years and to involve total funding in the order of £7M (selected from the options totalling £15M).

The programme is primarily aimed at extracting benefits from work within the UK. The study was therefore to concentrate on existing relevant work, or ideas of interest within the UK. No specific effort in the study was to be expended in seeking information on US programmes. There was a strong interest in capitalising on the ARE experience gained from existing data fusion, situation assessment and resource allocation work, and extending this to deal with the requirements of strategic and tactical missile defence and the higher functional levels of command and control.

The study has made some initial assumptions on the balance of activities and their time horizon. Three types of research effort were considered:

[1] It is based on ARE's Specification (Bolton 1989), Statement of Work (SDIO 1989), the SANDERLING Project Plan and Study Working Paper 1.

1. Applications

   This is driven primarily by the need to prove the application of a technology in a specific domain and would typically result in a demonstrator[2] or a specific prototype[3] .

2. Enabling

   This concerns itself with establishing generic theories and tools in support of other types of research. Different knowledge representation and reasoning methods fall into this category, which also includes a wide range of prototypes.

3. Speculative

   This is almost entirely unconstrained by near-term considerations of practical implementation, though it is still conducted with particular goals in mind. In the current context one might hypothesise radically different solutions and examine data fusion in that context.

Each of the above has a role to play, depending on the overall objectives of the programme. In particular, the balance between enabling and applications research can have a great impact on programmes beyond the projected research phase. It is often the case that applications research cannot directly be taken further in a related application area, or sometimes in the same area, because of simplifying assumptions made in the research.

For this study it was assumed that enabling research would be included, but with the proviso that it be directed towards Naval and SDI C^2 requirements. Applications work leading to demonstrators was also to be included, where such elements could be justified by the benefits of demonstrating techniques or technology within the timescale of the programme. If more speculative research was to be included, its relevance must be identified and related to an operational need.

The study was to take account of the objectives and status of existing and planned ARE projects in the field, including the TDS and TDP. Funds for the CCRP will include a large portion of ARE's research expenditure in this field over the next three years. Existing ARE commitments to projects were therefore to be reflected in the proposed research programme.

[2] For this study, 'demonstrator' implies a reasonably large scale, engineered and integrated system that starts to approach an operational system in complexity, scenarios etc.

[3] A 'prototype' implies an exploratory piece of practical experimentation (generally software). It is however based on particular functional problems in the application domain.

ARE experience has indicated that much of the research needs to be iterative, learning from previous projects and adjusting the future programme to respond to the lessons learnt. The proposed projects and programme should allow for this by indentifying appropriate dependencies and reviews.

It is intended that the CCRP complements a programme of demonstrator development. These demonstrators will provide a testbed for ideas generated during the research programme and/or for experiments. The CCRP should therefore include projects that make use of available demonstrators for carrying out experiments.

The study was asked to consider, but not to be restricted by, six specific areas for research: parallel computer architectures, human-computer interaction, validation and specification, database-KBS interfaces, planning and operational TMD applications.

The main SDI assumption has been that the primary interest and scenario driver for this work is Theatre Missile rather than Strategic Missile Defence. Thus the main input to the study has been from UK generated TMD documents. However, because of the limitation of these documents and their associated scenarios, the study has also taken some note of strategic defence issues from other documents.

It is planned that the CCRP runs in parallel with the latter phases of the UK Architecture Study (UKAS). The SANDERLING study should therefore complement UKAS, especially in the areas of discrimination and architecture specification.

2.2 Goals

The stated aim of the programme in the Statement of Work (SDIO 1989) is "The execution of a vigorous research programme to explore various opportunities to exploit the advantages knowledge-based systems techniques can provide in dynamic and complex defensive operations".

Within this aim the research programme should:

- research the enabling technologies that will support the further development of the TDS into an operational system;
- extend the scope of the current work into the higher levels of C^2;
- extend the applicability of the current work into the TMD domain.

To complement this aim and to help define objectives for the research programme, the study postulated technical goals, working assumptions and constraints about the:

- function of systems to be developed;
- performance criteria they must fulfil;
- users they must support;

- systems environment in which they will work;
- special operational requirements.

The following sections describe a set of such goals, for each of the Naval and TMD application areas. These goals were partly derived from existing work at ARE and partly from the anticipated or potential role for knowledge-based C^2 decision support systems.[4]

Ideally such goals should contain a strong element of future operational requirements and be driven primarily by user needs. Also, the sets of goals for research in both Naval and SDI fields should be co-ordinated, consistent, unambiguous and achievable. In practice, however, short time-frame research programmes are as much driven by incremental changes in technology and existing systems as they are by explicit operational goals. Nevertheless the process of defining goals has been a help in defining programme objectives.

The goals for the Naval and SDI components of the CCRP will be somewhat different. In the former there is a large corpus of existing work that has led to viable techniques for elements of the C^2 problem, and an existing infrastructure into which new systems must fit. In the latter, work takes place on a "greenfield site", there are few constraints imposed by existing systems and methods. However, many of these goals are common to both Naval and TMD requirements, for example those relating to situation assessment and the development of verification and validation techniques.

The following two sections describe an initial set of working goals, for each of the Naval and SDI application areas.

2.2.1 Naval Programme Goals

The following describes an outline requirement for a series of systems, exploiting a range of technologies, and with the overall aim of improving C^2 performance. A number of assumptions have been made about what should be included in such a requirement from functional and operational points of view. A number of questions are raised, for which research might provide answers.

Naval Command and Control has been functionally decomposed by ARE in AXT3 Technical Note No. 28 into the processes of:

- Data Fusion

  This deals with the processing of sensor signals and non-real-time data for tactical picture compilation. It handles the integration of data from disparate sources, of varying quality, to produce hypotheses identifying vehicles (ships, aircraft, etc.) and their allegiance.

- Situation Assessment

[4] For further detail see Working Paper 1, on which this section is based.

As opposed to the identification and classification processes of Data Fusion, Situation Assessment begins to draw inferences about the intentions of each vehicle, group of vehicles and the implications. An impression of the threat posed by any vehicle/group (or their friendly allegiance and role) is drawn.

- Resource Allocation

  In reaction to threats, or developing situations, a plan is composed and resources are committed to its implementation. This may need to be changed dynamically over a relatively short time period.

- Planning

  In order that appropriate resources are available at the right time for tactical and strategic offensive and defensive purposes, a higher level of planning activity takes place.

The detailed character of these functions depends quite heavily on the state of alert pertaining at any time. It is assumed that systems must operate appropriately at all alert states.

These functional goals have to be the primary aim of research effort but there are subsidiary objectives that impart other technical goals or constrain the way research effort should be applied towards functional goals. These further goals are dealt with in the sections below on performance, users, the systems environment and operational concerns.

Progress in research has led to the expectation that decision support aids performing at least the Data Fusion function could be deployed using straightforward developments of the present technology. This position itself leads to goals of two kinds, those relating to the operational exploitation of Data Fusion technology and those aiming to extend the technology base into the higher level functions. For the time being the former goal can be simply stated as:

"To rationalise the techniques used for the Data Fusion demonstrators to allow practical exploitation."

For the higher levels of C^2, however, the desired functionality can be expanded. The AXT3 Technical Note Number 28, "A Description of Data Fusion, Situation Assessment and Resource Allocation for knowledge-based Naval C^2" does precisely this, providing a useful decomposition of functional requirements for the initial three levels of the C^2 breakdown.

Within each functional level there is a need to examine specific problems that arise in naval practice, such as information saturation and hypothesis fixation for which the Data Fusion demonstrators offer potential solutions. This is intended to ensure that operational benefits accrue from the use of decision support systems.

2.2.1.1 Performance

Some of the basic performance requirements relate to the C^2 system interfaces (eg incoming data and user requirements). Firstly, there is an estimate that the number of input track reports to the Data Fusion level is in the order of 100 per second. From knowledge of the current Data Fusion capabilities this will equate to an input rate at the Situation Assessment level of around one significant change per second in the situation picture. Similarly, further estimates can be produced as events propagate through the various levels.

It is also known that objects may stay within the tactical picture for varying lengths of time, and that data will be accumulating from each sensor report. Given an estimate of the total number of tracks, further judgements can be made on the amount and nature of data storage that needs to be maintained on- or off-line.

At the other end of C^2 system there is a user. The usefulness of the system will be directly related to the timeliness of the information it delivers to the user. The maximum delay in throughput from sensor report to output display is dependent on the event to which the data relates. Many objects, including aircraft, will be tracked over several minutes, so delays of a minute may not be critical. Others, however, are far more fast-moving and second, or sub-second, responses have to be achieved.

These performance requirements will shape the research that needs to be done on implementation methods, particularly processing power requirements and real-time architecture considerations. However, they will also impact on the quality and type of information that can be made available to the user. Perversely, some of the most critical decisions will have to be made under such time pressure that high quality information cannot be produced quickly enough. Ways of presenting the available data in the most decision-effective manner therefore need research.

Aside from such considerations are factors such as the robustness of the hardware and software to partial and total failures. For example, is "graceful degradation" feasible and desirable and, if so, how is it achieved?

2.2.1.2 Users

Many aspects of the User Interface to the decision support system are tied up in the functionality requirements; the method and degree of user interaction and user's potential contribution to the reasoning process being examples.

A substantial issue to the users is the operational structure into which systems must fit and the need to reach a good match between the functions performed by the machines and the potential set of users of the machines. There already exist operational structures performing the C^2 function in the Naval context. The aids produced must either fit this structure, or propose and support a viable replacement.

In some instances operational command structures will be physically distributed, i.e. there may be several users at separate sites. Under these circumstances, issues of consistency between the pictures and advice offered at each site become important.

2.2.1.3 Systems Environment

Considerations here act more as constraints than goals : they restrict the number of valid solutions.

It is known, for example, that procurement cycles dictate that the performance of computing equipment being put into service at any time will lag that available in the best of commercial systems by a factor of around 10. Given projections of desired in-service dates and performance requirements this helps dictate the kind of hardware approaches that will be viable, in particular the choice between a purpose built processor and an "off-the-shelf" one, or the need for parallel processing.

It is also possible to estimate the bandwidth supported by projected system architectures for local (intra-ship) and remote (inter-ship) communication, and the timeliness of information that these links will deliver. Though integration with sensor data sources will be a component of TDS work, a hierarchical command structure, supported by the decision aids to be researched, would require distributing functions between systems with a consequent communications need.

2.2.1.4 Operational Concerns

Each of the layers of the C^2 system makes use of information produced by lower layers. This means that the programme must : (a) a build a clear understanding of the data exchanges for research at the higher levels to proceed, and ; (b) include research that supports the progressive introduction of the technology.

One of the most important constraints on the balance of research effort between enabling, applications and speculative projects is the timescale over which results are expected. The table below makes estimates of dates for the introduction of each level of functionality as a demonstrator and as a component of operational system requirements.

Decision Support Level	Demonstration	Requirement
Data Fusion	1991	1995
Situation Assessment	1995	1998
Resource Allocation	2000	2003
Battle Planning	2003	2005

Since it is expected that this research will lead to the procurement and in-service use of C^2 systems, the goals or constraints imparted by such requirements must be considered:

- Procurement/Systems Engineering

  Numerous issues arise here, including: how are large knowledge-based systems to be specified, validated, and integrated with other systems (including databases)? How is the development process for them to be managed?

- In-Service

  Similarly, what facilities should systems contain to support user training, exercises, maintenance and in-use modification/programmability?

2.2.2 SDI Programme Goals

The objective of the research programme from the SDI perspective is "To investigate and determine the utility of knowledge-based systems techniques and technology for BMC^2".

The main SDI assumption has been that the primary interest and scenario driver for this work is Theatre Missile rather than Strategic Missile Defence. Thus, the main input to the study has been from UK generated TMD documents. However, because of the limitation of these documents and their associated scenarios in the wider context, the study has also taken note of Strategic Defence (SD) issues and used available documents (SDS 1989).

The BMC^3 study (BMC^3 1987) has already gone some way into performing a top-down examination of the functional requirements stemming from the UK Architecture Study (Phase 6 Final Report). The authors identify six top-level activities:

- compile surveillance picture
- command BMC^2 system
- evaluate situation
- decide action
- manage BMC^2 sub-systems
- direct sensors and weapons.

Unlike the Naval domain, more of these processes are expected, at least in the TMD context, to be largely automatic. In the SD context a high degree of man-in-the-loop is required.

Numerous difficulties at the Data Fusion level have been identified, e.g. there are very many objects and they are clustered into unresolvable groups, there are many different sensor types, and their view is dependent on aspect angle, there is incompleteness and uncertainty in sensor data and there is clutter in the field of view. It is assumed that many of these problems will be examined by existing Data Fusion programmes. Similarly, considerations such as sensor differences between Naval and SDI cases should be considered within that work.

Areas already identified for research are:

- Track Correlation

  Because several sensors will be looking at very dense fields of objects from differing perspectives, correlating reports is a major difficulty. The problem is more complicated than the Naval problem in having to deal with 3D rather than $2^1/_2$D space, with all the origin registration problems that this imposes.

- Object Discrimination

  Assumptions are made that the re-entry vehicles (RVs) will be heavily disguised and that discrimination of them will require combining many sources of information. Information sources will have large amounts of uncertainty associated with them.

  Impact prediction can be done with some accuracy for each track. Associated with this is a resource management function that should be able to assess the effects multiple penetrations will have on the ability to respond.

2.2.2.1 Performance

The TMD scenarios project many times the number of objects in the sensor picture that exist in the Naval case. There are expected to be up to 80,000 independent objects (including decoys) identifiable in the field of view. A scenario is expected to evolve over a period as short as 400 seconds.

Given these approximate figures, it could be expected that the processing demand will be four to five orders of magnitude greater than for the Naval scenarios, at least at the Data Fusion level.

It should be stressed, however, that the relative magnitude of processing required diminishes at the higher function levels, since the tactical complexities of the TMD scenario are likely to be less than those pertaining in Naval warfare. Two orders of magnitude more processing might be expected at the Situation Assessment level, due to sheer numbers, and similar or better at the resource allocation level.

2.2.2.2 Users

Despite the expectation that the TMD system will largely function autonomously, it is inevitable that there will be users of the system who must assess the conclusions of its analysis and sanction responses. It is unlikely that a user would supplement or intervene in much of the low-level decision making since it is too time-critical. The requirement therefore is largely for rapid and accurate situation display and high level, high speed, decision support.

2.2.2.3 Systems Architecture

The architecture studies into Theatre Missile Defence have proposed a three tier system, within which decision support aids will have to operate :

- Sector
- Regional
- Theatre

The functions to be performed within this architecture mean that there will be considerable data transfer between nodes, since both data and commands are moving between hierarchy levels. Within the system architecture considerations, tolerance of the network to loss of nodes is of considerable importance. Each node is to have a high degree of autonomy and the co-ordination of functions across the network is therefore worthy of study.

2.2.2.4 Operational Considerations

The scope of the CCRP covers the "mid-term threat" identified by UKAS for 2005-2015. It is assumed that an on-line demonstrator would therefore be a goal for 2005.

Issues of procurement and in-service use of systems correlate closely with the requirements for Naval systems.

2.3 Objectives

The goals discussed in section 2.2 were used to derive three levels of objectives for the programme.

- Level One : the major objectives of the programme;
- Level Two : the specific objectives that support each top-level objective;
- Level Three : the specific research objectives to be achieved by individual projects or groups of projects.

Part A considers Levels One and Two objectives. Level Three objectives for each proposed research project are discussed in Part C .

2.3.1 Level One Objectives

Three top-level objectives for the research programme have been established:

- to advance the capability to deploy Naval KBS based C^2 systems at sea;
- to extend the functionality of current KBS based C^2 demonstrator systems;
- to provide the basis on which to deploy and deliver KBS based solutions to functional requirements for TMD C^2 systems.

The first objective is primarily focussed on the engineering, performance and development issues associated with moving the current technology out into operational systems at the earliest date. It also includes a strong element of user justification, benefit assessment and risk reduction. The time horizon for achieving this objective is short to medium term (1 to 6 years).

The second objective accepts that the capability of current technology to provide the full range of future functional requirements will be limited and that work is necessary to investigate and explore more advanced ideas and to incorporate these into future demonstrators. This objective has a medium to long term time horizon (6 years or more).

The third objective combines the two previous objectives but concentrates specifically on the TMD domain with a medium to long term time horizon (5 to 15 years)[5].

2.3.2 Level Two Objectives

Each of the Level 1 objectives has been expanded into a set of Level 2 objectives, which are discussed in this section, and provide more specific technical detail.

Input to this section has included ARE documents on the objectives of the TDS[6] and TDP[7]. TDS objectives are described in Byrne (1989a). TDP objectives are categorised under two headings: Development/Procurement ; and Experimental (Miles 1989a). The Experimental Programme Research Objectives (EXPROS) are amplified in Narborough-Hall (1989).

Where appropriate, cross reference is made to ARE objectives in each Part C project description. The extent projects meet these objectives is also used as one of the factors for evaluating whether or not a project should be included in the programme.

2.3.2.1 Deployment Capability

- To provide practical support to the TDS trials deployment via:
    - training and assistance in the experimental use of the TDS;
    - modifications for performance optimization;
    - evaluation and assessment of TDS capability;
    - data collection, processing and analysis.

[5] See Part B section 2.2 for a detailed definition of assumptions on time horizons.

[6] See Byrne (1989a)

[7] See 'TDP Experimental Programme Strategy', TDP/13.1/1 Issue B 12 Jan 90, C S Narborough-Hall for a recent summary of the documentary position on the TDP.

- To extrapolate TDS experience to operational system deployment by examining the feasibility of scale and performance extensions to meet future operational requirements.

- To make use of the TDS deployment to:

  - define KBS development methods;
  - establish metrics for assessing system performance;
  - assess the level of user/organisational impact;
  - refine the user requirements for advanced C^2 systems;
  - establish improved mechanisms for database/KBS interaction;
  - improve the way in which real time issues are accommodated;
  - determine the feasibility of the TDS design and systems architecture;
  - define a growth/development path for the TDS.

2.3.2.2 Enhanced Functionality

- To extend, via prototypes, the functional support provided by KBS further into non-Data Fusion areas such as Situation Assessment, Resource Allocation and Planning.

- To investigate and exploit new techniques for knowledge representation and manipulation.

- To establish the feasibility and benefits of parallel architectures for functions such as data fusion.

- To establish an outline architecture for an integrated battle management prototype.

- To investigate novel/non-KBS areas of significant potential advantage such as distributed AI, machine learning and neural networks.

2.3.2.3 TMD Requirements

- To explore and define solutions to development techniques in critical aspects such as:

  - verification, validation, specification and maintenance;
  - performance limitations and improvements;
  - significance of HCI issues.

- To extend, via prototypes, the functional support provided by KBS further into C^2 areas such as Situation Assessment, Resource Allocation and Planning.
- To investigate and discover new techniques for knowledge representation and manipulation.
- To establish the feasibility and benefits of parallel architectures for functions such as Data Fusion.
- To establish an outline architecture for an integrated battle management prototype.
- To investigate novel areas of significant potential advantage such as machine learning and neural networks.

3 STUDY APPROACH

3.1 Requirements

The approach adopted in the SANDERLING Study was aimed at defining a research programme which addresses the development goals of ARE and SDIO, as well as highlighting the opportunities for C^2 systems created by current developments in AI and KBS.

The requirements for the method were that it should :

- take into account ARE and SDIO requirements;
- set realistic research targets;
- create a balanced, costed research programme;
- generate an audit trail to justify the composition of the final programme.

3.2 Principles

The study method was based on the following principles :

3.2.1 The definition of objectives

The study method viewed the goals of the programme in terms of a hierarchy of objectives (see section 2.3), addressed in the following sequence :

Level One Objectives :	the high-level objectives of Naval and TMD C^2 - these are the operational characteristics which are required of these systems;
Level Two Objectives :	the specific objectives within each top-level objective;
Level Three Objectives :	the research objectives to be achieved - these are the objectives of the individual research projects.

3.2.2 Analysis of the current state of the art

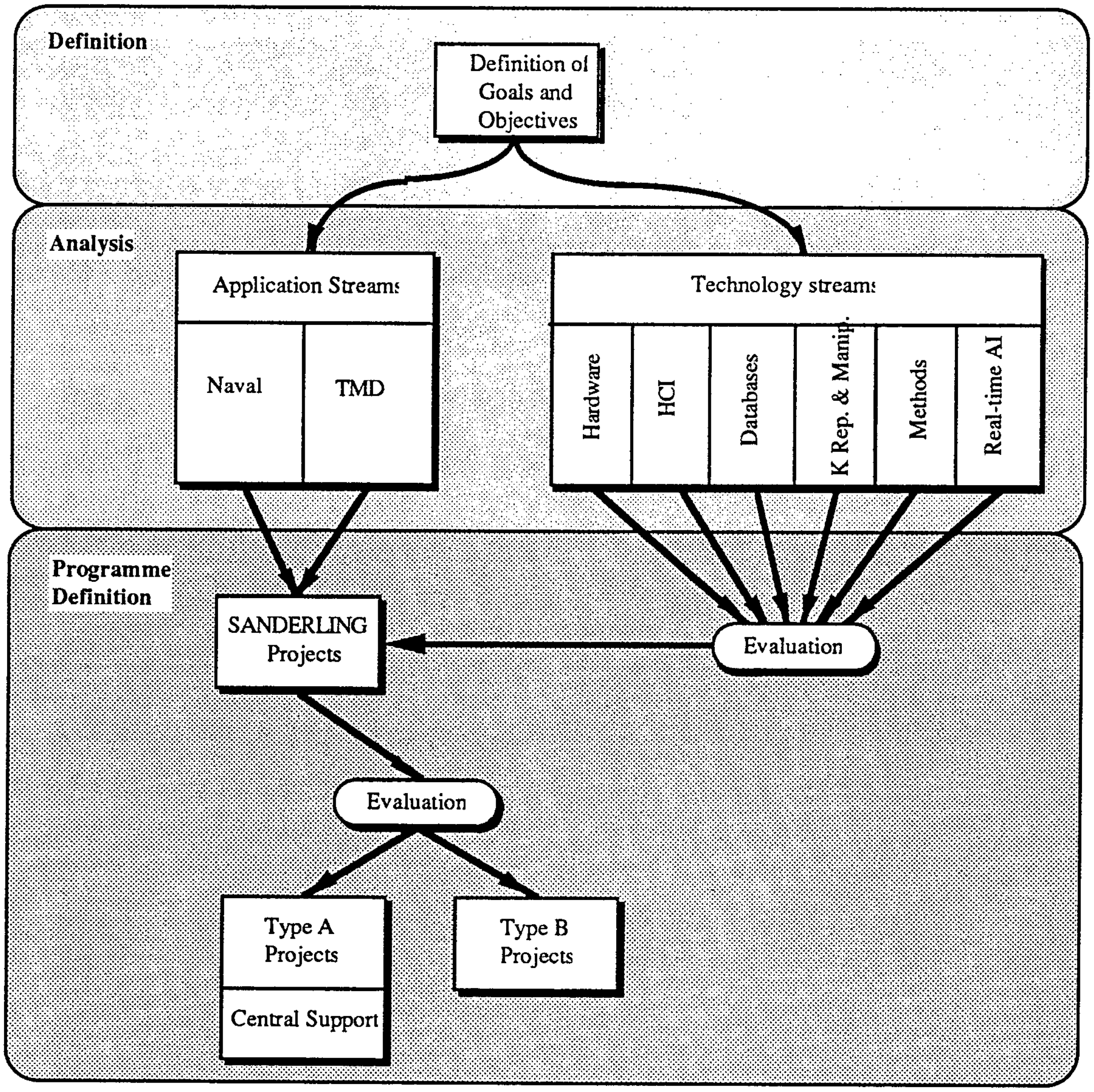
The assessment of the opportunities created by recent developments involved a detailed analysis of the state of the art in the relevant fields, with an emphasis on the level of appropriate research activity in the UK.

3.2.3 Three stage process

The study was carried out in three stages :

- Definition - establish the objectives.
- Analysis - explore a wide range of technical opportunities.
- Programme definition - focus down on a realistic programme of projects.

The relationship between stages is illustrated below. (For the explanation of Type A and Type B projects see Section 3.3.3)



3.2.4 Collaborative Working

The nature of the SANDERLING study demanded that the team should exploit a wide range of expertise in the various technical disciplines, whilst maintaining a tightly focussed view of the overall objectives of the study. This was achieved by adopting a different mode of working in each of the three stages of the study :

Stage 1 : Brainstorming

The initial definition stage of the study required a creative, free-ranging discussion of objectives and goals. For this reason, it was carried out by a series of intensive brainstorming sessions, involving most members of the full SANDERLING team.

Stage 2 : Parallel assessment of applications and relevant enabling technologies.

The second stage of the study required input from a large number of technical specialists, both internal and external to the team. In addition to addressing the immediate technical issues raised by the TDS / TMDD projects, the research programme explored the opportunities for C^2 created by recent developments in AI and KBS. It also reflected the relative importance of research within each of the enabling technologies. For these reasons, the study adopted a parallel assessment of applications and relevant enabling technologies. Each major technical area was assigned to a SANDERLING team member, who had the responsibility of exploring that technology, in consultation with research staff at ARE and external technical authorities.

The technology and application perspectives were brought together in the definition of the final programme.

Stage 3 : Core Team

The final stage of the study required a consolidation of the results of the technical analysis and a focussing on the application issues. For this stage a small core team was assembled from representatives of each of the three contractors. This team had the responsibility of co-ordinating the input from the other team members, and provided a forum for discussions on the technical direction of the study.

3.2.5 The evaluation of technologies and projects

The scope of the study was very broad. It covered a wide range of technical subjects and could easily have generated a very large number of potential research projects. For this reason, it was essential that a stringent evaluation procedure should be employed to ensure that the research programme contained only the most critical projects.

The evaluation process was applied both to the technologies and to the proposed projects. The aims of evaluating the technologies were twofold - to eliminate those which were peripheral to the main research thrust, and to give an early indication of the priorities which should be given to projects on the basis of their technical components.

The evaluation of the projects themselves was based on an assessment of :

- critical importance to the programme objectives;
- technical risk / timescales;

- compatibility with existing ARE programmes;
- availability of UK expertise.

3.2.6 Programme composition

The proposed programme defines a set of high priority research projects, the total cost of which was within the £7M budget limit, and a second set of lower priority projects taking the total up to twice the budget. The first set are the recommended set of projects for the programme. However, recognising that the criteria for selecting projects could change in the future, the Final Report includes a full set of project descriptions and costings, together with the tabulated justifications for the selection of projects.

Also included in the programme are :

- a provision for adequate central support facilities, eg scenario generation;
- estimates of time and effort;
- timetables and dependency details.

The aim in formulating the programme was to generate a vigorous and effective research programme, which achieves an appropriate balance between applied and enabling research and which exploits the maximum synergy between Naval and TMD projects.

3.3 Stages

3.3.1 Definition

The first stage in the study was aimed at defining the development goals of Naval and TMD C^2 and identifying the enabling technologies which are needed to support them. This included an understanding of the role of C^2 systems in these domains and the operational constraints under which they must perform. The definition process involved the following activities :

- a familiarisation study of developments in Naval and TMD C^2, including briefings from ARE and SDIPO and reading of reference material;
- a first brainstorming session to define the characteristics of the two domains, the goals and objectives of the research programme, and the engineering issues posed by the enabling technologies;
- a second brainstorming session to define the technology streams to be investigated in the following phase of the study.

The Definition stage generated a working paper (WP1) which set out the objectives and assumptions of the study, and a set of eight research streams for further investigation. Six of the streams addresses technical issues and are refered to as the technology streams :

- Hardware Architectures
- Human-Computer Interaction
- Database / Knowledgebase Interaction
- Development Methods
- Knowledge Representation & Manipulation
- Real-Time Systems Design & Distributed Artificial Intelligence

The remaining two streams were concerned with the application domains, and served to guide and focus the results from the other streams. These are the Application Streams :

- Naval Applications
- TMD Applications

3.3.2 Analysis

The aims of the analysis stage of the study were to identify the principal sub-topics within each enabling technology and to assess the relevance and importance of each sub-topic to the objectives of the research programme. This stage of the study was carried out as a parallel exercise, each stream being researched by a different member of the SANDERLING team. Appropriate team members were allocated on the basis of their background and expertise in the technical areas in question. The analysis involved the following :

- **Application Analysis**

  The analysis of the Application Streams was aimed at generating a more detailed understanding of the technical objectives of the two application areas. This was carried out by :

  - consultation with appropriate members of ARE's research team;
  - a study of the phase 5 and 6 UKAS reports;
  - drawing on background experience of C^2 applications.

- **Technical Analysis**

  Each Technology Stream was studied in depth by the appropriate team member. The technical analysis involved :

  - identification of the key technical issues for the two application areas, by consultation with members of ARE's research team, external experts and those SANDERLING team members concerned with the Application Streams;

  - a detailed study of the state of the art in the relevant technologies, by literature studies and consultation with acknowledged experts in the field;

  - identification of the principal centres of expertise in the UK.

- **Technical evaluation**

  The sub-topics within each Technology Stream were evaluated and ranked according to a set of criteria which assessed their criticality to the research objectives and the degree of risk involved in researching them. The results of the analysis and evaluation processes were reported in two working papers (WP2 & 3).

3.3.3 Programme Definition

The final research programme was derived from the results of the Analysis Stage. This was carried out as follows :

- **Identify projects**

  An initial set of research projects was generated by the Application Streams, based on the perceived needs of the Naval and TMD C^2 programmes. An attempt was made to achieve an appropriate balance between short-term, applied research and longer-term enabling research. Projects were classified into the following categories :

  - Category 1 : TDS Support

  - Category 2 : Applied Research

  - Category 3 : Enabling Research

  - Category 4 : Central Support

- **Analyse technical components**

  Each project was analysed in terms of the technical components which would need to be researched to achieve its goals. Individual technical components were identified and investigated in detail by the appropriate Technology Stream member.

- **Costing**

  The cost of each project was assessed by a combination of bottom-up costing of the individual technical components, and a top-down analysis of the scope of the project. Effort costings were based on a set of standard rates for commercial and university staff.

- **Evaluate and short-list**

  Each project was evaluated according to a set of criteria which assessed its critical importance to the programme objectives. The projects were then ranked within the principal categories and the research programme was assembled from these lists.

  Two lists of projects were defined : the Type A list of the highest ranking projects costing up to the budget limit of £7M, and the Type B list comprising the remainder, costing up to twice the budget limit.

- **Technical Review Panel**

  The full set of projects were reviewed by a panel of senior academic and industrial experts. Feedback from that review was incorporated into the final programme.

- **Assemble programme**

  In determining the composition of the programme, the following factors were taken into account :

  - the balance between applied and enabling research;
  - the balance between Naval / TMD research;
  - synergy between Naval and TMD research;
  - the provision of adequate central support facilities;
  - dependencies between projects;
  - relation to planned and on-going ARE research.

Annex A1 : Glossary of Artificial Intelligence and Computing Terms

Agent Modelling : is concerned with modelling actor's intentions and plans, and their modification in the light of information concerning an adversary's intelligence and plans (ie. adversarial game-playing).

Approximate Reasoning : (see Progressive Reasoning)

ATMS (Assumption-Based Truth Maintenance System) : is a type of belief-revision system developed by J. Doyle that is domain independent. Assumptions are propositions whose belief depends on the disbelief on other propositions.

Auto-epistemic (see Epistemic) : knowledge that includes information on its own structure, derivation and production.

Automated Theorem Proving : deducing logically correct statements from a set of axioms and other theorems using a model or program that is based on a mathematical system of logic.

B-tree : or "binary-tree" is a self organising storage mechanism that works by taking the necessary number of storage locations and building them into a tree structure to minimise the length of the access route.

Bayes (theory / statistics) : Bayes is school of statistical thought that is based upon the assignment of prior probabilities to a set of events in order to derive a figure for the likelihood of a consequent event. It is a type of inference that readily lends itself to implementation in KBS, but is not universally accepted.

Blackboard : a control mechanism often used in data-driven knowledge-based systems where the firing of a set of rules places facts in a storage place or 'blackboard', where they can activate other rules that use the same location for matching. Applications often require establishing a hierarchy of blackboards for different purposes.

Circumscription : involves the use of an axiom schema in a first-order language to express the idea that certain formulas have the smallest possible extensions consistent with the axioms. It is used as a way of representing common-sense assumptions about the world.

Closed World (logics / assumptions) : in closed world systems an assumption is made about the completeness of statements about the world. It is assumed that the system has access to all information that might result in statements being true, such that anything that is not shown to be true can be inferred to be false (rather than simply undefined).

Closure : the sentence obtained from a formula by attaching a universal quantifier to each of its free variables.

Combinatorial explosion : in many AI and search problems the amount of processing increases exponentially with the number of cases or instances. This is referred to as a 'combinatorial explosion' and results in the resources required to find the optimum solution becoming unaffordable as the problem is scaled-up.

Commit (1 to n-phased) : in the context of databases systems, a commit is a process which updates (usually to disk) a set of pending updates. Where a logical update spans more than one database or table, it is normal to introduce phases into the commit process so that each component of the update can be signalled and, in the case of the failure of one component, the rest of the transaction can be backed out and re-tried.

Concurrency Handling : in database systems it is often the case that more than one process or user is attempting to update a piece of information at the same time. To deal with this a 'concurrency handling' mechanism is implemented to manage the processes that are in contention and ensure that both retain an accurate view of the data.

Conflict Resolution : in knowledge-based systems, once a set of rules has been evaluated to true, the inference engine has to determine what the order of precedence should be. This involves a conflict resolution strategy.

Constraint Propagation : a type of problem or reasoning process whereby paths to possible solutions are generated by iteratively applying a set of constraints to points in the search space until an admissable solution is found.

Contracting / negotiating systems : a control mechanmism that is modelled on the commercial model of packets of work being put out to tender to bodies that bid to do it if they are fit and have the available resource. It is intended to find the most appropriate process for a piece of work, and may involve numerous levels of sub-contracting depending on the implementation.

Data Manipulation Language : in a database management system, a DML is the syntax used for populating and managing the database and is used by the system for accessing data. It may or may not be the same as the Query Language, although there are clear advantages to there being consistency between the two.

Data-flow : a parallel processing architecture in which processing components are pipelined together depending on their data requirements to enable concurrency to be exploited.

Deduction : the reasoning process whereby given a set of initial conditions and a system of logic, a person or system is able to produce a set of new, logically correct statements about the world.

Deep reasoning : some cognitive scientists maintain a distinction between deep and shallow reasoning, where the former involves an understanding of underlying causality, whereas the latter may involve simple heuristics, eg. 'press the accelerator to go faster' pre-supposes no knowledge of the underlying workings of a car.

Default logic : makes assumptions about a logical system, such that if a particular statement is not proven then it is taken to be false, or other mechanism provided for assigning it a status other than simply unknown.

Distributed Artificial Intelligence (DAI) : concentrates on modelling decision making involving a number of agents that may or may not be physically remote. Central issues are the delegation of control, handling of component failure, communication, and organisational models for multi-agent decisions.

Domain specific : a domain is an area of expertise. Many types of decision-making and heuristic knowledge appear to be very specific in their applicability, and has led a number of KBS practitioners to focus on domain-specfic models of thought.

Epistemic Reasoning : reasoning about the structure, derivation and reliability of knowledge.

Epistemological : relating to the theory of knowledge. Epistemology concerns the basis upon which we believe we know things and the way in which we organise our knowledge.

Normal Form : based on Codd's theory of data - 1st Normal Form eliminates repeating groups of attributes for a single key; 2nd. Normal Form separates attributes which depend on a part of the key and not the whole of it; and Third Normal Form eliminates transitive dependencies.

First Order Predicate Logic / Calculus : a logical system, defined by Frege, in which variables can only range over objects, rather than permitting quantification over variable functions and predicates (known as Omega Order logics).

Focus of attention : in reasoning systems, it is often necessary to direct the inferencing process such that new facts are produced in a structured or coherent way, rather than the system pursuing different lines of reasoning in a way that humans find erratic.

Formal Methods : these are methods for the production of computer software from mathematically provable specifications.

Frame : a frame is a flexible storage structure that enables information about a class of objects to be stored. Different frames can be used to store a hierarchy of information of increasing specificity. In object-oriented programming frames also support the use of 'message-passing' via procedural attachments.

Fuzzy Logic : as implemented in knowledge-based system, involves a mechanism for dealing with rules of the sort 'If X then the probability of Y is 0.3'. Systems of this type require careful interpretation and checking to ensure that assumptions about the independence of events are valid.

Garbage Collection (incremental) : in dynamic programming languages (as often used in knowledge-based systems) there is reliance upon run-time binding and in-memory data-structures. Memory gradually fills ups with structures that are no longer required and the system must then remove these or 'garbage collect'. This may be done using a 'stop and sweep' strategy or while the main processing continues, using an 'incremental' strategy.

Global data / hypotheses : in knowledge-based systems the 'facts' or knowledge are often accessible to all other parts of the system, acting as a global data store of information about a putative world.

Heuristic (Search) : a heuristic is a rule-of-thumb. Heuristics are often used by humans to arrive quicky at solutions to problems, that are either cannot be solved optimally, or are very large. Additionally, heuristics are often a good representation of how people rationalise their decision-making behaviour.

Hypothetico-deductive (model / reasoning) : is a model of human reasoning or scientific progress, whereby progress is made by the postulation of new theories and the generation of tests or experiments that can refute or support those theories. The results of the experiments may result in a re-formulation of the original hypothesis.

Inference Engine : the component of Knowledge Base System that controls the sequence of the evaluation and firing of rules, managing their effect on the knowledge base of facts.

Interrupt (synchronous / asynchronous) : in real-time systems, an interrupt is an event that causes an alteration in the performance of the system. The event may be allowed to occur at any time (asynchronous) or be controlled by the system or external co-ordinated system (synchronous).

Introspective consistency : in a logical or reasoning system the consistency of all statements with each other and with the logical system for deducing new statements or maintaining truth content. In human terms, this refers to the co-herence of someone's reasoning. The content with respect to the external world is not questioned.

Knowledge Acquisition : this is the process of getting knowledge from a human expert. The knowledge may or may not be placed in some form of intermediate representation, before being turned into software.

Knowledge Representation Language : this is a paper or software system for encapsulating human knowledge in such a way that it is accessible and operable on by inference components. The KRL may be specific to a particular field of expertise, or general in nature.

Lazy Evaluation : in LISP and some logical expressions, the minimal expansion of lower-level terms so that they fulfil certain data or success requirements, rather than progressing exhaustively.

Locking : in database management systems, there are times when more than one user or process requires to access or modify the same data. To handle this contention, a 'locking' strategy is used, such that one process cannot update a data resource that is being used by another process. Such transactions may be backed-out or re-tried depending on the strategy being used.

Machine Learning : the study of computational methods for acquiring new knowledge, new skills and new ways of organising existing knowledge. Central areas include learning by example and learning by analogy.

Meta-level : the level above the one being considered. Often used in the context of control systems where meta-level knowledge refers to knowledge about what the system knows and how it can modify its behaviour.

Methods (see Object Oriented) : methods are part of the object-oriented paradigm and are the procedural attachments. A method is activated by a message being passed to it and then acts, often upon data hidden within the object. Methods can be inherited from higher levels in an object hierarchy.

Modal logics : logics that are based on ideas of 'necessity' and 'possibility'. It is sometimes taken to include any logic that is not based on predicate and propositional logic.

Model-theoretic (semantics) : based on a theory that uses a formal mathematical model of the world in order to set up relationships between linguistic elements and their meanings.

Multiple Worlds : there are many problems that involve generating alternative pathways and then comparing the information at each of the resulting nodes. These are often referred to as hypothetical reasoning or multiple worlds problems. Software to support such problems must store state information and allow control to pass throughout the search space depending on a strategy such as best-first, or depth-first.

Neural Network : a model of pattern-matching and discrimination that has its origins in the construction of models of human neuronal firing, but has now extended beyond this. Decisions are arrived at by the taking of a number of inputs into a set of layers of nodes that behave deterministically given certain inputs. The behaviour of the network is 'taught' by the presentation of a set of example cases.

Non-monotonic : refers to a reasoning process or logical system, where information that is true in a given context, may subsequently become untrue, and require the retraction of a set of statements for which it formed the basis. In a monotonic system, by contrast, something that is once true, remains true.

Object Oriented (databases / languages) : is a software development paradigm and style that comprises a number of components : a dynamic programming environment (delayed binding); data hiding (objects can only operate on their own data); message passing (parameter passing to cause actions); and inheritance structures (modelling hierarchies of data and associated methods).

Ontology : the study of being, in the context of logical or knowledge-based systems the philosophical basis of objects or components in models.

Optimistic Locking (see **Locking**) : is a database locking strategy where the likelihood of contention is felt to be small and updates may be lost. This may happen in a real-time environment where, rather than an update transaction locking a resource, it checks at the end to see whether another transaction has updated the data since the beginning of the transaction and, if it has, is backed out.

Paging : most computer systems require more working memory than is physically available in the machine. To get around this, and to make applications more portable across machines of differing sizes, data is 'paged' or transferred from primary memory (RAM) to secondary memory (disk). The term paging is used as the unit of transfer is often known as a page.

Parallelism : is a hardware or software term that refers to the splitting of a set of operations into a set of tasks that can be executed at the same time. These tasks may take place on different processors.

Persistence : in dynamic programming languages and environments data structures or objects may be created in working memory. Persistence refers to the longevity of these structures through their committing to secondary memory.

Prior Probabilities (see **Bayes**)

Procedural Attachements (see **Methods**)

Production System : in knowledge-based systems is the paradigm of having a set of facts in working memory, a set of rules and an inference engine that controls the firing of those rules. The system is said to 'produce' new facts when the rules fire.

Progressive Reasoning : in order to support time-constrained processing it is possible to employ a control mechanism that guarantees responses within certain periods of time at the expense of completeness. With such a strategy the quality of the responses increases with available time.

Query Processor : is the component of a database management system that takes an end-user or query-language query and generates the necessary data access statement and plan. It shields the user from questions over the efficiency of different access routes, by taking over this function.

Reason Maintenance: in a knowledge-based or logic system, it is often helpful to store information about why certain decisions or branches in the processing were taken. This is referred to as reason maintenance.

Reified (epistemic logic) : in a logical system, the reduction of higher order terms to First Order Predicate Logic (see First Order Predicate Logic).

Repertory Grids : a knowledge acquisition technique based upon Kelly's Personal Construct Theory, which classifies entities, usually on the basis of the difference an individual ascribes between similar and dissimilar examples of each entity.

Resolution : a logic programming technique of inference that begins with a pair of clauses that are unifiable (can be made identical by some permitted substitution of terms) and then 'resolves upon' them. The technique can always derive explicit contradictions from an inconsistent set of clauses.

RETE systems : in a forward chaining inference system a significant proportion of processsing time can be spent evaluating the right had side of rules to see if they are true. A RETE system is a network that stores the changes that occur between cycles of the inference engine (ie. each time a rule fires) so that the amount of evaluation can be minimised.

Rule Induction : a rule-induction system is one in which the system is presented with a number of test cases and the system then produces a set or rules using an algorithm that discriminates the key characteristics of the data.

Semantic Networks : a Semantic network is a structure (usually in software) that is used to represent a set of concepts and their inter-relation. Such conceptual structures have been adopted by cognitive scientists in an attempt to pursue the goal of emulating human thought.

Semantic validation (database) : rather than just ensuring that the data is of the correct type and length, semantic validation involves checking against other data or against application-specific information.

Situated Action : a view of modelling decision processes that places emphasis of the context of people's actions as being of prime importance in shaping their perception and expectations of the outcomes.

Situation Calculus : a logical notation expressing the key dimensions of a given real-world state, facilitating their comparison and logical manipulation.

Space Reclamation (see Garbage Collection) : in a dynamic programming environment, memory gradually becomes filled up with data structures for which there may no longer be any use.

Symbolic Processing : a generic name for a philosophy and style of programming that treats the symbols being manipulated by the system as representations of some external reality rather than merely flat data. The school of thought has been responsible for Knowledge-Based Systems and other systems that seek to replicate or emulate human thought processes.

Toolkit : in knowledge-based systems a 'toolkit' is a piece of software comprising a number of different functional components (eg. providing production rules and truth maintenance) that are packaged as an integral whole. The degree of seamlessness of the different parts is a good indication of a toolkit's maturity.

Tractable : manageable, in the sense of being capable of solution with some ease, as opposed to logically soluable but computationally unacceptable solutions (eg. NP-complete problems).

Transitive Closures : given a relationship between entities of the same type, it is the set of all instances that result from the exhaustive application of that relationship to its domain.

Truth Maintenance : is a logical or computer system that seeks to ensure that a set of logical relations or statements that are defined to hold are enforced when new facts are asserted by an inference component, thereby ensuring the logical consistency of the system.

Tuple : a row in a relational table. A relational table is one which has been reduced to Third Normal Form or 'normalised'.

Unary (operator) : having only one - in the case of an operator, one argument.

Validation : ensuring the system is satisfactory in terms of a set of success criteria for its use in a given context.

Verification : ensuring that a piece of software satisfies a specification of what it should do functionally.

Von Neumann machine / architecture : is the computer architecture that has been dominant since the invention of computers of having a serial memory into which consecutive statements are loaded and executed one at a time, as opposed to the splitting of the code into components which can occur in parallel, possibly on different hardware platforms.

Well-formed (sentence, formula - WFF) : is one which conforms to the axioms of the logical or semantic system to which it is meant to belong and is therefore of an acceptable form, if not truth content.

Annex A2 : References

To save repetition, all works referred to in the SANDERLING documents are listed here, arranged alphabetically by author and date.

Addison, E.R., Cain, M., Desmond, W. (1988) Criteria for Choosing Connectionist Paradigms for Real-Time Data Fusion and Adaptive Discrimination. Neural Networks, 1 (Supplement 1), 42, Pergamon Press.

Adey, S. (1987) Planning for a Dynamic World. Submissions to the 7th Alvey Planning SIG Workshop, IEE 1987.

AGARD (1987) The practical assessment of pilot workload. Advisory Group for Aerospace Research & Development, AGARDograph No. 282, June.

AI Magazine (1989) Issue on Universal Planning. Vol. 10, No. 4, (Winter).

Allen, J. (1981) A general model of action and time. TR 97, Dept of Computer Science, University of Rochester, Rochester, NY.

Allen, J. (1982) Maintaining knowledge about temporal intervals. Communications of the ACM, 26, 832-843.

Allport, D., and Kidd, A.L. (1989) Using Knowledge about Search Spaces to give Co-operative Responses. Detailed Abstract, Hewlett Packard.

Ambros-Ingerson, J.A., Steel, S. (1988) Integrating Planning, Execution Monitoring . AAAI-88.

Anderson, A.R., Belnap, N.D. (1975) Entailment. Vol 1, Princeton University Press.

Anderson, B, Lakin, W.L., and Byrne, C.D. (1989) Operational Use of Knowledge-Based Military C^2 Systems. Proceedings 3rd International Conference C^3MIS, IEE, London pp 43-47.

Anderson, J.M. (1988) The integration of HCI principles in structured system design methods. Proc. of MILCOMP.

Andre, E., Bosch, G., Herzog, G. & Rist, T. (1986) Characterizing Trajectories of Moving Objects Using Natural Language Descriptions. ECAI 86

Appelt, D. (1982) Planning natural-language utterances to Satisfy multiple goals. Technical Report TN-256, SRI International.

ARE staff (1990) TDS Level 0 Milestones. ARE Document, TDS/MP/1/3 Issue 1,9 March 1990.

Artificial Intelligence (1989) Special Issue on Genetic Algorithms (May).

Avignon 89 = Ninth International Workshop on Expert Systems and their Applications. EC2 1989.

Ayache, N. and Faugeras, O.D. (1988) Building, registering, and fusing noisy visual maps. International Journal of Robotics Research, 7(6) pp45-65

Bacharach, M. (1975) Group decisions in the face of differences of opinion. Management Science, 22 pp182-191.

Bacharach, M. (1979) Normal bayesian dialogues. J. American Statistical Soc., 74 pp837-846,

Ballim, A., and Wilks, Y. (1989) Artificial Believers. In preparation.

Bar-Shalom, Y. (1978) Tracking methods in a multi-target environment. IEEE Trans. Automatic Control, 23(4) pp 618-626.

Bar-Shalom, Y. (1981) On the track to track correlation problem. IEEE Trans. Automatic Control, 25(8) pp 802-807.

Bar-Shalom, Y. and Fortmann, T.E. (1988) Tracking and Data Association. Academic Press.

Bell, M., Bennett, M.E., and Zanconato, R. (1987) CCL Final Report C1924-FR-001a

Bennett, M.E. (1985) Resource Allocation with Scripts. Proceedings 3rd Alvey Planning SIG Workshop, IEE 1985.

Berger, J. (1985) Statistical Decisions (second edition). Springer Verlag, Berlin, GDR.

Bernsen, N.O. (1989) General introduction: A European perspective on cognitive science. In, A. Baddeley and N.O. Bernsen (Eds), Cognitive Psychology, Lawrence Erlbaum Assoc,. P.viii.

Bertin, A., Tornielli, G., de Ravinel, B. (1989) Diagnostic Reasoning with the QUIC Toolkit. ESPRIT '89 Conference Proceedings, Kluwer Academic Publishers.

Blackwell, A.F. (1988), Spatial Reasoning for Robots: A Qualitative Approach. MSc Thesis, Victoria University of Wellington, 1988.

BMC3 (1987) High Level Functional Structure for European BMC3. SS/1065/W1/136/040387/YY, 4th. March 1987.

Bolle, R.M. and Cooper, D.B. (1986) On optimally combining pieces of information, with application to estimating 3-d complex-object position from range data. IEEE Trans. Pattern Analysis and Machine Intelligence, 8(5), pp. 619-638, 1986.

Bolton, D.J. (1989) Specification for KBS Research Programme Study. ARE Paper AXT 7.50.0, May 1989.

Bond, A.H. and Gasser, L. (1988). Readings in Distributed Artificial Intelligence. Morgan Kaufmann, 1988.

Boolos G.S. and Jeffrey, R.C. (1980) Computability and Logic. Cambridge University Press (2nd. edition).

Boose, J. H. (1989) A Survey of Knowledge Acquisition Techniques and Tools. International Journal of Knowledge Acquisition. Vol. 1, No. 1. pp. 3 - 37.

Box, G.E.P. and Jenkins, G.M. (1976) Time Series Analysis: Forecasting and Control. Holden-Day, Oakland, California.

Bratman, M. (1986) Intentions, plans and practical reason . In preparation.

Brodie, M.L., Balzer, R., Weiderhold, G., Brachman, R., and Myopoulos, J. (1986) Knowledge Base Management Systems : Discussions from the Working Group. In Kerschberg (ed.) (1986) Expert Database Systems, Benjamin Cummings Publishing

Brown, C.M. and Durrant-Whyte, H., Leonard, J. and Rao, B. (1989) Centralized and decentralized kalman filter techniques for tracking, navigation and control. In Proc. DARPA Image Understanding Workshop, pp 651-676. Morgan Kaufmann Inc.

Bruten, J. (1989) An Experiment in the Use of a Deontic Logic of Actions for the Specification of a Configuration Knowledge Base. Hewlett Packard.

Bundy, A. (1985) Incidence Calculus: A Mechanism for Probabilistic Reasoning. Proceedings of Workshop on Uncertainty in Artificial Intelligence, UCLA, Los Angeles, 1985.

Burton, M and Shadbolt, N. (1989) Knowledge Engineering. Technical Report 87-2-1.

Byrne, C.D. (1989) Objectives of Technology Demonstrator Programme. ARE Document TDS/16.1/1 Issue 2, 6 September..

Byrne, C.D., Miles, J.A.H., and Lakin, W.L., (1989) Towards Future Knowledge-Based Naval Command Systems. Proceedings 3rd International Conference C^3MIS, IEE, London pp 33-42.

Byrne, C.D. (1990) ARE AXT(R) Research Project Priorities. ARE Document ARE/23.01.02/90 Issue, 8/9 February 1990.

Cammarata, S. and McArthur, D. (1983) Strategies of cooperation in distributed problem solving. In Int. Joint Conf. Artificial Intelligence, p. 767

Carbonell, J.G. (1981) Counterplanning: A Strategy Based Model of Adversary Planning in Real-World Situations. Artificial Intelligence, Vol 16, pp295-329.

Card, S.K., Moran, T.P and Newell, A. (1983) The psychology of human-computer interaction. Lawrence Erlbaum Associates.

Carroll, J.M. (1987) Interfacing thought. Cognitive aspects of human-computer interaction. The MIT Press.

Cavanna, A. et al (1989) QUIC Toolkit Demonstrator Applications. ESPRIT '89 Conference Proceedings, Kluwer Academic Publishers.

Chandrekaran, B. (1981) Natural and social system metaphors for distributed problem solving. IEEE Trans. Systems Man and Cybernetics, 11:1.

Charniak, E., McDermott, D.V. (1985) Introduction to Artificial Intelligence. Addison-Wesley 1985.

Chellas, B. (1980) Modal logic. Cambridge: Cambridge University Press.

Chong, C. (1979) Hierarchical estimation. In 2nd MIT/ONR CCC Workshop, Monterey CA, 1979.

Chong, C., Chang, K., and Bar-Shalom, Y. (1986) Joint probabilistic data association in distributed sensor networks. IEEE Trans. Automatic Control, 31(10) pp889-897.

Cleal, D.M., Heaton, N.O. (1988) Knowledge-based systems: implications for human-computer interfaces. Ellis Horwood.

Cohen, P.R. (1985), Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach. Pitman 1985

Cohen, P.R., Levsque, H.J. (1987) Persistence, Intention and Commitment , in Reasoning about actions and plans. Kaufmann

Collins, G.E. (1975) Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition ,Proceedings 2nd GI conference on Automata Theory and Formal Languages. Lecture Notes on Computer Science 33, Springer 1975.

Conry, S.E., Meyer, R.A., Lesser, V.R. Multistage Negotiation in Distributed Planning. Readings in Distributed Artificial Intelligence (Edited by Bond and Gasser)

Daniel, J., (1989) Engineering Knowledge Based Data Fusion Systems: Solving the Run-time Bottleneck. Proceedings of the International Conference on the Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, Tennessee.

Davis, R., Smith, R.G. (1983) Negotiation as a metaphor for distributed problem solving. Artificial Intelligence, 20 p63.

de Kleer, J. (1986) An assumption based TMS. Artificial Intelligence, 28(2) pp127-162

de Kleer, J. (1986) An Assumption-based Truth Maintenance System. Artificial Intelligence, Vol 28, 1986, pp127-162.

de Kleer, J. (1988) A General Labelling Algorithm for Assumption-based Truth Maintenance. AAAI-88.

de Kleer, J. (1989) A Comparison of ATMS and CSP Techniques. IJCAI-89.

Dean, T. (1985) Temporal Imagery: An Approach to Reasoning about Time for Planning and Problem Solving . Tech Report YALE/CSD/RR # 433.

Dean, T., Boddy, M. (1988) An Analysis of Time-Dependent Planning. AAAI-88.

DeGroot, M.H. (1974) Reaching a consensus. J. of the American Statistical Assoc., 64(345) pp 118-121.

Dixon, M., de Kleer, J. (1988) Massively Parallel Assumption-based Truth Maintenance. AAAI-88.

Donald, B.R. (1988) A geometric Approach to Error Detection and Recovery for Robot Motion Planning with Uncertainty , Artificial Intelligence, Special Volume on Geometric Reasoning, Vol 37, Nos 1-3, December 1988.

Downing, K.L. (1987) Diagnostic Improvement through Qualitative Sensitivity Analysis and Aggregation , AAAI-87, pp789-793.

Doyle, J. (1979) A Truth Maintenance System , Artificial Intelligence, Vol 12, 1979, pp231-272.

Doyle, J., McDermott, D. (1980) Non-Monotonic Logic I , Artificial Intelligence, Vol 13, 1980.

Dubois, D., Prade, H. (1988), Default reasoning and Possibility Theory , Artificial Intelligence, vol.35, 1988

Durfee, E.H., Lesser, V.R., Corkill, D.D. (1987) Cooperation through communication in a distributed problem solving network. In M.N. Huhns (1987) editor, Distributed Artificial Intelligence. Morgan Kaufmann.

Durfee, E.H., Lesser, V.R., Corkill, D.D. (1985) Coherent Co-operation Among Communicating Problem Solvers , Technical Report 87-07, department of Computer and Information Science, University of Massachusetts, Amherst, 1985.

Durrant-Whyte, H.F. (1987a) Consistent integration and propagation of disparate Sensor information. Int. J. Robotics Research, 6(3) p24.

Durrant-Whyte, H.F. (1987b) Integration, Coordination, and Control of Multi-Sensor Robot Systems. Kluwer Academic Press, Boston, MA.

Durrant-Whyte, H.F. (1988a) Sensor models and multi-sensor integration. Int. J. Robotics Research, 7(6) pp97-113.

Durrant-Whyte, H.F. (1988b) Uncertain geometry in robotics. IEEE J. Robotics and Automation, 4(1) pp23-31.

Durrant-Whyte, H.F., Rao, B.Y., Hu, T. (1989) Toward a fully decentralised architecture for multi-sensor data-fusion. In IARP Int. Conf on Multi-Sensor Data Fusion, 1989.

EFD (1986) Making the Most of an Incomplete Picture - Command Strategies for Assessing Ambiguous and Unreliable Information , Electronic Facilities Design Report ref: EFD/859/FR03B, Contract: NSW32A/1117, August 1986.

EFD (1988) Force Command Decision Making and Information Flow in ASWEX 2/87 , Electronic Facilities Design Report ref: EFD/8701/FR02.1B, Contract: NSW32A/1597, June 1988.

EFD (1988b) The ASWC in an ASWEX Scenario: Roles and Responsibilities ASWEX 1/88 , Electronic Facilities Design Report ref: EFD/8701/FR04.3B, Contract: NSW32A/1597, December 1988.

EFD (1989) An Analysis of the use of the Force ASW Plot , Electronic Facilities Design Report ref: EFD/8701/FR04.2B, Contract: NSW32A/1597, February 1989.

EFD (1989b) Decision Making at the Level of the ASW Strike Force Command , Electronic Facilities Design Report ref: EFD/8701/FR05.1A, Contract: NSW32A/1597, March 1989.

Engelmore, R., Morgan, T. (1988) Blackboard Systems , Addison-Wesley.

Etherington, D.W. (1988) Reasoning with Incomplete Information, Morgan Kaufmann, 1988

Evans, J.St.B.T. (1982) The psychology of deductive reasoning: Routledge and Kegan Paul.

Evans, J.St.B.T. (1989a) Bias in human reasoning: causes and consequences: Lawrence Erlbaum Assoc.

Evans, J.St.B.T. (1989b) Problem solving, reasoning and decision making: in, A. Baddeley and N.O. Bernsen (Eds), Cognitive Psychology, Lawrence Erlbaum Assoc,.85-102.

Fagin, R., Halpern, R. (1985) Belief, awareness, and limited reasoning: Preliminary report. IJCAI-85, 491-501.

Faugeras, O., Ayache, N. (1986) Building visual maps by combining noisy stereo measurements. In Proc. IEEE Int. Conf. Robotics and Automation, pp 1433-1438, San Francisco, U.S.A.

Firby, R., McDermott, D. (1988) Representing and Solving Temporal Planning Problems . In N. Cercone and G. McCalla (eds) The Knowledge Frontier, Springer-Verlag.

Fleck, M.M. (1987) Representing Space For Practical Reasoning , IJCAI 87,

Flynn, A.M. (1985) Redundant sensors for mobile robot navigation. Master's thesis, MIT, 1985.

Forbus, K.D. (1983) Qualitative Reasoning about Space and Motion in (ed. D. Gentner & A.Stevens) Mental Models, Lawrence Earlbaum 1983

Forgy, C.L. (1982) RETE: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artificial Intelligence 19, no. 1.

Foster, I., Gregory, S., Burt, A.D., Ringwood.G.A. (1989) An Abstract Machine for the Implementation of PARLOG on Uniprocessors. In New Generation Computing Vol. 6 Pt. 4, pp 389-420.

Fox, M. S. (1983) Techniques for Sensor-Based Diagnosis. Proc. IJCAI 83.

Fox, M.S., Lowenfeld, S., Kleinosky, P. (1983) Techniques for sensor-based diagnosis. Technical Report CMU-RI-TR-83-7, The Robotics Institute, Carnegie Mellon University.

Frisch, A.M., Allen, J.F. (1989) Knowledge Retrieval as Limited Inference, in Myopoulos and Brodie (1989).

Gallaire, H., Minker, J., Nicolas, J.M. (1989) Logic and Databases : a Deductive Approach, in Myopoulos and Brodie (1989).

Galliers, J.R. (1989) A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-agent Conflict , University of Cambridge Computer Laboratory, Technical Report 172.

Gardarin, G., Valduriez, P. (1989) Relational Databases and Knowledge Bases, Addison-Wesley

Gardenfors, P. (1988) Knowledge in Flux: Modelling the Dynamics of Epistemic States , Bradford Books, MIT Press, 1988.

Gardin, F., Meltzer, B. (1989) Analogical Representations of Naive Physics , Artificial Intelligence, vol.38, no.2, March 1989

Garey, M., and Johnson, D.S. (1979) A Guide to the Theory of NP-Completeness, Freeman.

Garvey, M.A., Jackson, M.S. (1989) Introduction to object-oriented databases, Information and Software Technology, Vol 31 no. 10, 1989 pp. 521-528.

Garvey, T., Lowrance, J. (1981) An inference technique for integrating knowledge from disparate sources. In Proc. International Joint Conference on Artificial Intelligence, p319.

Gasser, L., Braganga, C., Herman, N. (1987) MACE: a flexible testbed of distributed AI research. In M.N. Huhns, editor, Distributed Artificial Intelligence. Morgan Kaufman.

Gelb, M. (1974) Applied Optimal Estimation. MIT Press.

Genesereth, M.R., Ginsberg , M.L., Rosenschein, J.S. Co-operation Without Communication , Readings in Distributed Artificial Intelligence (Edited by Bond and Gasser)

Georgeff, M.P., Ingrand, F.F. (1989) Decision making in an Embedded Reasoning System , IJCAI-89.

Georgeff, M.P., Lansky, A.L. (1987) Reactive Reasoning and Planning , AAAI-87.

Gilmore, J. (1989) Planning in an autonomous system environment , Avignon 89.

Ginsberg, M. (1989) A Circumscriptive Theorem Prover , Artificial Intelligence, vol.39,no.2, June 1989

Giralt, G. (1984) Research trends in decisional and multi-sensory aspects of third generational robots. In Second Int. Symp. Robotics Research, Koyoto, Japan. MIT Press.

Goguen, J. (1988) Modular Algebraic Specification of some Basic Geometrical Constructions , Artificial Intelligence, Special Volume on Geometric Reasoning, Vol 37, Nos 1-3, December 1988.

Goodstein, L.P., Andersen, H.B., Olsen, S.E. (1988) Tasks, errors and mental models: Taylor & Francis.

Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Gordon, J., Shortliffe, E.H. (1985) A Method For Managing Evidential Reasoning in a Hierarchical Hypothesis Space , Artificial Intelligence, vol.26,no.3,July 1985.

Gregory, S. (1987) Parallel Logic Programming in PARLOG - The Language and its Implementation. Addison Wesley.

Gupta, A. (1987) Parallelism in Production Systems. Pitman

Hager, G. (1987) Information maps for active sensor control. Technical Report MSCIS-87-07, U. of Pennsylvania Dept. Comp. Sci., 1987.

Hager, G. (1988) Active Reduction of Uncertainty in Multi-Sensor Systems. PhD thesis, University of Pennsylvania.

Hager, G., Durrant-Whyte, H.F. (1988) Information and multi-sensor coordination. In Uncertainty in Artificial Intelligence 2, pp 381-394. North Holland.

Hager, G. (1984) Modeling Mental Models: Experiments in Cognitive Modeling of Spatial Reasoning , ECAI 84, Elsevier Science Publishers

Halpern, J. (ed.) (1986) Theoretical Aspects of Reasoning about Knowledge Morgan Kaufmann.

Halpern, J.Y., Rabin, M.O. (1987) A Logic to Reason about Likelihood , Artificial Intelligence, vol 32, October 1987

Harmon, P. (ed.). Expert Systems Strategies.Knowledge Acquisition. Vol. 5, No. 7.

Harmon, S.Y. (1986) Sensor data fusion through a blackboard. In Proc. IEEE Int. Conf. Robotics and Automation, p 1449.

Harmon, S.Y. Gage, D.W., Aviles, W.A., Bianchini, G.L. (1984) Coordination of intelligent subsystems in complex robots. In Proc. IEEE Conf. Applications of AI, p 64.

Harris, C.J. (1988) Reducing Uncertainty and Ignorance in C2 Systems in Harris, C.J. Application of Artificial Intelligence to Command and Control Systems, Peter Peregrinus, 1988.

Harris, C.J., White, I. (1987) Advances in Command, Control and Communication Systems. IEE Press.

Hashemipour, H.R., Roy, S., Laub, A.J. (1988) Decentralized structures for parallel kalman filtering. IEEE Trans. Automatic Control, 33(1), pp 88-93.

Hashimoto, O., Miyai, H. (1987) A user interface prototyping tool: Proceedings of HCI'87, September.

Haugh J.M., (1989) Formal Software Development Techniques Applied to Expert System Production, Proceedings of the 6th IASTED International. Symposium EXPERT SYSTEMS: Theory and Applications, Long Beach, California, 14-16 December 1989.

Haugh, J.M., (1988) The Formal Specification and Investigation of a Blackboard System Knowledge-base ARE Technical Memorandum AXT88020

Hautin, F., Vailly, A. (1986) La coopération entre systemes experts. Journees nationales du PRC/IA, Cepadues-Editions.

Hayes, P. (1977) In defence of logic . IJCAI-5, pp559-565.

Hayes-Roth, B. Hayes-Roth, F. (1979) A Cognitive Model of Planning , Cognitive Science, Vol 3, pp275-310, 1979.

Heath, T. (1990) TDS Software Issues, ARE Document TDS/M1/1 Issue 1, 23 March 1990.

Henderson, D.A. (1986) The TRILLIUM user interface design environment: CHI'86 Proceedings, ACM Press, April

Henderson, T. (1987) Workshop on multi-sensor integration. Technical Report UUCS87-006, U. Utah Computer Science.

Henderson, T., Hansen, C. (1985) The Specification of distributed sensing and control. J. of Robotic Systems, 2(4) pp387-396 .

Henderson, T., Fai, W., Hansen, C. (1984) Mks: A multisensor kernel system. IEEE Trans. Systems Man and Cybernetics, 14(5) pp784-791.

Henderson, T., Weitz, E., Hansen, C., Michie, A. (1988) Multisensor knowledge systems: Interpreting 3-D structure. Int. J. Robotics Research, 7(6) pp114-137.

Hewitt, C.E. (1986) Offices Are Open Systems , ACM Transactions of Office Information Systems, April 1986.

Hillman, D.J. (1985) Artificial intelligence: Human Factors, 27, 1, pp21-31

Hintikka, J. (1962) Knowledge and belief . Cornell University Press.

Hintikka, J. (1969) Semantics for propositional attitudes . In J. Davis et al. (eds) Philosophical Logic.V Dordrecht: Reidel.

Hirst, R.A., (1989) Allegiance Assessment Using Explanation-Based Reasoning Proceedings 3rd International Conference C^3MIS, IEE, London pp 56-63.

Ho, Y.C. (1980) Team decision theory and information structures. Proceedings of the IEEE, 68 p644.

Ho, Y.C., Chu, K.C. (1972) Team decision theory and information structures in optimal control. IEEE Trans. Automatic Control, 17 p15.

Holman, R. (1987) High Level Human Factors Requirements for Surface Naval Command Systems. ARE TN (AXC) 87009.

Holman, R., Turley, N., Harvey, R.S. (1987) Human Performance Requirements and Specification for a Surface Naval Command Systems. ARE TN (AXC) 87011 (UK Restricted).

Holman, R., Turley, N. (1988) Methods for Integrating Human Factors into the Early Development of Surface Naval Command Systems. ARE TN (AXC) 88001.

Hudlicka, E., Lesser, V. R. (1984) Meta-level Control Through Fault Detection and Diagnosis. Proc. AAAI-84.

Hunter, J.R.W. (1986) Qualitative Models in Medicine: An Artificial Intelligence Perspective , AIMG-10, AI in Medicine Group, School of Engineering and Applied Sciences, University of Sussex, 1986.

HUSAT Research Centre (1988) Human factors guidelines for the design of computer-based systems:

Ingalls, D., Wallace, S., Chow, Y., Ludolph, F., Doyle, K. (1988) Fabrik: A visual programming environment; OOPSLA'88 Proceedings, ACM Press, September.

Israel, D., Brachman, R. (1981) Distinctions and Confusions: A Catalogue Raisonee , IJCAI-81.

Johnson-Laird, P.N. (1983) Mental models; Cambridge University Press.

Kaelbling, L.P. (1986) An Architecture for Intelligent Reactive Systems, Proc. of 1986 Workshop on Reasoning about Actions and Plans, Morgan Kaufmann.

Kapur, D., Mundy, J.L. (eds.) (1988) Artificial Intelligence: Special Volume on Geometric Reasoning, vol.37, nos. 1-3, December 1988

Kidd, A.L. (1989) Deriving Principles for Generating Co-operative Responses - An Experiment , Hewlett Packard.

Klein, D., Finin, T. (1987) What's in a Deep Model?: A Characterisation of Knowledge Depth in Intelligent Safety Systems , IJCAI-1987, pp559-562.

Klein, G.A.. (1990) Recognition Based Decision, to appear in Advances in Man-Machine Systems Research, Vol. 5, ed. W.R. Rouse.

Konolige, K. (1986) A deduction model of belief . London: Pitman.

Korf, R.E. (1987) Real-Time Heuristic Search: First Results , AAAI 87.

Kornfeld, W.A., Hewitt, C.E. (1981) The Scientific Community Metaphor , IEEE Transactions on Systems, Man and Cybernetics, January 1981.

Krijgsman, A.J., Verbruggen, H.B., Bruijn, P.M. (1988) Knowledge Based Real-Time Control Proceedings IFAC Conference on Artificial Intelligence in Real-Time Control, Wales, 1988.

Kuczewski, R,M. (1987) Neural Network Approaches to Multi-Traget Tracking. In Proceedings of the First IEEE Int. Conf. on Neural Networks IV, pp 619-633.

Kuipers, B. (1983), Modeling Knowledge of Routes: Partial Knowledge and Individual Variation , AAAI 83,

Kuipers, B. (1986) Qualitative Simulation , Artificial Intelligence, Vol 29, No 3, September 1986.

Kuipers, B. (1987) Reasoning, Causal Encyclopaedia of AI, Shapiro, S.C. (ed), Wiley, 1987.

Laffey, T.J., Cox, P.A., Schmidt, J.L., Kao, S.M., Read, J.Y. (1988) Real-time Knowledge-based systems, AI Magazine, Spring 1988 pp 27-43.

Laird, J.E., Newell, A. and Rosenbloom, P.S. (1987) Universal Subgoaling and Chunking: The Automatic Generation and Learning of Goal Hierarchies (Kluwer)

Le Naour, P., Desjouis, M. (1989) SEDAIR: Un Systeme 'Aide de Camp' Pour L'Exploitation de L'Information Orchidee , Avignon 89

Lenat, D.B., Hayes-Roth, F., Klahr, P. (1979) Cognitive Economy , Stanford Heuristic Programming Project HPP-79-15, June 1979.

Lesser, V.R., Corkill, D.D. (1979) The application of artificial intelligence techniques to cooperative distributed processing. In Proc. IJCAI, pp 537-540.

Lesser, V.R., Corkill, D.D. (1981) Functionally accurate, cooperative distributed systems. IEEE Trans. Systems Man and Cybernetics, 11(1) pp81-96.

Lesser, V.R., Pavlin, J., Durfee, E. (1988) Approximate Processing in Real-Time Problem Solving , AI Magazine Spring 1988.

Lin, F. (1987) Reasoning in the presence of inconsistency , AAAI 87, Morgan Kaufmann

Lingard, A., Baker, M., Davidson, R., Hook, M. (1988) The Command Support Tool: A Naval Tactical Planning Support Environment , Report of the 8th workshop of the Planning Special Interest Group, Nottingham, IEE 1988.

Marcus, S. SALT. IJMMS.

Markosian, L.Z., Rockmore, A.J. (1984) A Paradigm for Dynamic Plan Generation and Execution Monitoring , Proceeding of the 3rd Seminar on Application of Machine Intelligence to Defence Systems, RSRE 1984.

Marr, D., Nishihara, H.K. (1977) Representation and Recognition of the Spatial Organisation of 3-D Shapes , Proc. Royal Society B, Vol 200, 1977, pp 269-294.

Marroquin, J.L. (1985) Probabilistic solution of inverse problems. Technical Report TR-860, MIT Artificial Intelligence Laboratory.

Marshak, J. and Radnor, R. (1972) The Economic Theory of Teams. Yale University Press, 1972.

Martins, J. and Shapiro, S. (1988) A model for belief revision. Artificial Intelligence, 35(1) pp 25-79.

Martins, J.P. & Shapiro, S.C. (1988) A Model for Belief Revision. Artificial Intelligence,vol.35, no.1, May 1988

McCarthy, J., Hayes, P. (1969) Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie (eds) Machine Intelligence 4 , Edinburgh: Edinburgh University Press.

McDermott, D. (1982) A temporal logic for reasoning about processes and plans. Cognitive Science, 6,101-55.

McDermott, D. (1987) AI, Logic and the Frame Problem , In F. Brown (ed.) The Frame Problem in AI , Morgan Kaufmann.

Miles, J.A.H (1988) Artificial Intelligence Applied to Data Fusion and Situation Assessment for Command And Control. PhD Thesis, Faculty of Engineering and Science, Southampton University.

Miles, J.A.H. (1989a) Objectives of Technology Demonstrator Programme, ARE Document TDP/1.1/1 Issue 1, 31st October 1989.

Miles, J.A.H. (1989b) Technology Demonstrator System (TDS): Technical Requirements Specification. ARE AXT3 Technical Note Number 90 (UK Restricted)

Miles, J.A.H., (1989) Architectures for C^2 Knowledge-Based Systems Proceedings 3rd International Conference C^3MIS, IEE, London pp 64-69.

Miller, D. (1985) Planning by Search through Simulations. Tech Report YALE/CSD/RR # 423.

Ministry of Defence (1984-89) Defence Standard 00-25 (Parts 1-12). Human factors for designers of equipment; MOD Directorate of Standardization.

Miranker, D. (1989) TREAT: A New and Efficient Match Algorithm for AI Production Systems. Pitman.

Mitche, A. and Aggarwal, J.K. (1986) Multiple Sensor integration through image processing: A review. Optical Engineering, 23(2) p380.

Molyneux, S.D., (1990) TDS HCI Requirements Specification, ARE Technical Report TDP/3.1/2.

Molyneux, S.D., Montgomery, J.D. (1990) The User Interface Design Specification, ARE Technical Report TDP/7.2/1.

Montgomery, A. (1988) GEMINI - Government Expert Systems Methodology Initiative. Research and Development in Expert Systems V. pp. 14 - 24.

Montgomery, J.D., Byrne, C.D., (1989) Evolving the User Interface for Knowledge-Based Command and Control Systems Proceedings 3rd International Conference C^3MIS, IEE, London pp 48-55.

Moon, J.R., Marriette, J.P., Frampton, S.P. (1988) Uncertainty in Command and Control Systems , in Harris, C.J. Application of Artificial Intelligence to Command and Control Systems, Peter Peregrinus, 1988.

Moore, R. (1984) The role of logic in Artificial Intelligence . SRI Technical Report, 335.

Moore, R. (1985) A formal theory of knowledge and action , in Hobbs and Moore (eds) Formal theories of the common sense world . Norwood, N.J.: Ablex.

Moore, R. (1988) Autoepistemic logic. In P. Smets, E. Mamdani, D. Dubois and H. Prade (eds.) Non-standard logics for automated reasoning London: Academic Press.

Moore, R., Hendrix, G. (1979) Computational models of belief and the semantics of belief sentences . SRI Technical Note 187.

Morris, P.A. (1977) Combining expert judgments: A bayesian approach. Management Science, 23(7) pp 679-693.

Myopoulos, J., Brodie, M. (eds.) (1989) Artificial Intelligence and Databases, Morgan Kaufmann, San Mateo, Ca.

Nandhakumar, N., Aggarwal, J. (1986) Integrating information from thermal and visual images for scene analysis. In Proc. SPIE Conf. Applications of Artificial Intelligence, pp 132-142, 1986.

Narborough-Hall, C.S., TDP Experimental Programme Strategy. ARE Technical Report TDP/13.1/1, Issue A.

Nash, J.F. (1950) The bargaining problem. Econometrica, p155,1950.

Ng, K., Abramson, B. (1989) A Model for using Multiple Knowledge Bases in an Expert System, Avignon 89.

Nii, H.P. (1986) Blackboard Systems. AI Magazine, 1986.

Nii, H.P., Aiello, N., Rice, J. (1988) Frameworks for Concurrent Problem Solving: A Report on CAGE and Poligon in R. Engelmore, T Morgan Blackboard Systems , Addison-Wesley 1988.

Nilsson, N.J. (1986) Probabilistic Logic , Artificial Intelligence, Vol 28, No 1, Feb 1986.

Nilsson, N.J. (1989) On 'Logical Foundations of Artificial Intelligence', a response to the reviews by S. Smoliar and J. Sowa , Artificial Intelligence, Vol 38, No 1, Feb 1989.

Orlando, N.E. (1984) An intelligent robotics control scheme. In American Control Conference, p204.

Osga, G.A. (1989) User-computer interface issues for future ship combat consoles; in, Proceedings of the Human Factors Society 33rd Annual Meeting, 1079-1083.

Ow, P.S., Smith, S.F., Thiriez, A. (1988) Reactive Plan Revision , AAAI-88.

Pearl, J. (1986) Fusion, Propagation and Structuring in Belief Networks , Artificial Intelligence,vol.29, no.3, September 1986

Porrill, J., Pollard, S.B., Mayhew, J.E.W. (1987) Optimal combination of multiple sensors including stereo vision. Image and Vision Computing, 5(2) pp174-180.

Price, C., Hunt, J. (1989) Qualitative Reasoning to build Diagnostic Expert Systems in Shadbolt, N. Research and Development in Expert Systems VI , Cambridge University Press 1989.

Priest, G. (1989) Reasoning about Truth Artificial Intelligence, vol.39, no.2, June 1989

Provan, G.M. (1989) An Analysis of ATMS-based Techniques for Computing Dempster-Shafer Belief Functions , IJCAI 89

Przymusinski, T.C. (1989) An Algorithm to Compute Circumscription , Artificial Intelligence, vol.38, no.1, February 1989

Raimondo, A. (1989) Analyse d'Intentions Tactiques , Proceedings Avignon '89 Expert Systems and their Applications , May 1989.

Rao, B.S.Y., Durrant-Whyte, H.F. (1989) A fully decentralized algorithm for multisensor kalman filtering. Technical Report 1787/89, Oxford U. Robotics Research Group

Rapaport, W. (1986) Logical foundations for belief representation. Cognitive Science, 10, 371-422.

Rasmussen, J. (1976) Outlines of a hybrid model of the process plant operator; in, T.B. Sheridan and G. Johannsen (Eds), Monitoring Behaviour and Supervisory Control, Plenum Press, 371-384.

Rasmussen, J. (1986a) Information processing and human-machine interaction: An approach to cognitive engineering; North Holland.

Rasmussen, J. (1986b) A cognitive engineering approach to the modelling of decision making and its organisation; Riso National Laboratory, Report No. Riso-M-2589, June.

Reichgelt, H. (1989) Logics for reasoning about knowledge and belief.

Reinfrank, M., Dressler, O. Brewka, G. (1989) On the Relation Between Truth Maintenance and Autoepistemic Logic , IJCAI 89

Reiter, R., de Kleer, J. (1987) Foundations of Assumption-based Truth Maintenance Systems, AAAI 87, Morgan Kaufmann

Rescher, N. (1975) Plausible reasoning , Amsterdam: Van Gorcum 1976.

Rice, J. (1989) The Advanced Architectures Project AI Magazine Winter 1989.

Rice, J. (1989b) The ELINT Application on POLIGON: The Architecture and Performance of a Concurrent Blackboard System , IJCAI-89.

Roberts, M.E.C., Hughes, T.J. (1989) Uncertainty in situation assessment for pilot and instructor advisory systems , Avignon 89, EC2

Rosenschein, J.S. (1982) Synchronization of multi-agent plans. In American AI Conference, p115.

Rosenschein, J.S., Genesereth, M.R. (1985) Deals Among Rational Agents , Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 1985.

Sacerdoti, E. (1975) A Structure for Plans and Behaviour , Technical Report #109, SRI.

Salle, S.E., Årsen, K-E. (1989) A Comparison between Three Development Tools for Real-Time Expert Systems: Chronos, G2 and MUSE , Proceedings IEEE Workshop on Computer Aided Control Systems, Tampa, Florida, December 1989.

Salvendy, G. (1987) Advances in Human Factors/Ergonomics, 10B. Cognitive engineering in the design of human-computer interaction and expert systems. Proc. of 2nd Int. Conf. on HCI, 10-14 August; Elsevier Science.

Schlecter, T.M., Togila, M.P. (1986) New directions in cognitive science; Ablex Publishing Corp.

Schreiner,F., Pesa, G. (1987) A Parallel Architecture for Production Systems. University of Kaiserslauten.

SDIO (1989) Statement of Work for KBS Research and Demonstrator, 21 March 1989.

SDS Phase 1 System Description Document, (UK Release) F144-23-2, 1st. Dec. 1989.

Shafer, G., Logan, R. (1987) Implementing Dempster's Rule for Hierarchical Evidence , Artificial Intelligence, vol.33, 1987

Shafer, S.S., Thorpe, C.E. (1986) An architecture for sensor fusion in a mobile robot. In Proc. IEEE Int. Conf. Robotics and Automation, pp 2002-2011.

Shneiderman, B (1987) Designing the user interface: strategies for effective human-computer interaction; Addison-Wesley.

Singer, R.A., Kanyuck, A.J. (1971) Computer control of multiple site correlation. Automatica, 7(6) pp455-463.

Skingle, B. (1987) The Validation of Knowledge Based Systems. Proc. KBS 87. pp. 27 - 36.

Sloman, A. (1985) Real-Time Multiple-Motive Expert Systems , Proceedings BCS Expert Systems Conference, 1985.

Smith, R.G. (1980) The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver , IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-10, no. 12, December 1980.

Smith, R.G., Davis, R. (1981) Frameworks for cooperation in distributed problem solving. IEEE Trans. Systems Man and Cybernetics, 11 p61.

Smith, S.L., Mosier, J.N. (1986) Guidelines for designing user interface software: The MITRE Corporation, Report No. MTR 10090, ESD-TR-86-728, August.

SOAR: An architecture for general intelligence; Artificial Intelligence, 33, 1-64.

Stefik, M. (1981) Planning and Meta-Planning (MOLGEN: Part 2) , Artificial Intelligence 16 (1981).

Stiny, G. (1980) Introduction to Shape and Shape grammars , Environment and Planning B, Vol 7, 1980.

Suchman, L.A. (1987) Plans and Situated Actions Cambridge University Press 1987.

Suwa, M., Scott, A. C., Shortliffe, E. H. (1982) An Approach to Verifying Completeness and Consistency in a Rule-based Expert System. AI Magazine, Fall 82.

Tate, A. (1976) Project planning using a hierarchical non-linear planner. Dept of AI Report 25. Edinburgh University.

Tate, A. (1985) A Review of Knowledge Based Planning Techniques , BCS Knowledge Engineering Review, Vol 1, No 2, June 1985.

Tobat, D.L. , Rogers, S.K., Ross, S.E. (1986) Sentinel: An Expert System Decision Aid for a Command, Control and Communications Operator , SPIE Vol 365 Applications of Artificial Intelligence III, 1986

Trehan, R. (1989) Concurrent Logic Languages for the Design and Implementation of Parallel AI Systems. Dept of AI Paper No. 427. Edinburgh University.

Tsang, E. (1987b) TLP - A Temporal Planner. AISB-87

Turner, R. (1984) Logics for Artificial Intelligence . Chichester: Ellis Horwood.

Tychonievich, L., Zaret, D., Mantegna, J., Evans, R., Muehle, E., Martin, S. (1989) A Manoeuvring-Board Approach to Path Planning with Moving Obstacles , IJCAI 89

van Benthem, J. (1982) The logic of time. Dordrecht: Reidel.

Vardi, M. (ed.) (1988) Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge. Los Altos, CA: Morgan Kaufmann.

Vere, S. (1983) Planning in Time: Windows and Durations for Activities and Goals. IEEE Trans on Patt Recog and MI 246-267.

Walters, J.R., Nielsen, N.R. (1988) Crafting Knowledge Based Systems , Wiley 1988.

Wason, P.C., Johnson-Laird, P.N. (1972) Psychology of reasoning: structure and content; Batsford.

Weening, J.S. (1989) Parallel Execution of Lisp Programs. Stanford University report STAN-CS-89-1265

Weerahandi, S., Zidek, J.V. (1981) Multi-bayesian statistical decision theory. J. Royal Statistical Soc. A, 144(1) pp85-93.

Weerahandi, S., Zidek, J.V. (1983) Elements of multi-bayesian decision theory. The Annals of Statistics, 11(4) pp1032-1046.

Wesson, R., Hayes-Roth, F. (1980) Network Structures for Distributed Situation Assessment RAND report R-2560-ARPA, August 1980.

Whittington, G., Spracklen, C.T. (1989) Applications of Neural Networks to Tactical and Sensory Data Fusion Problem First IEE Conference on Artificial Neural Networks, London.

Wickens, C.D. (1984) Engineering psychology and human performance; Charles Merrill.

Williges, B.H., Williges, R.C. (1984) Dialogue design considerations for interactive computer systems; in, F.A. Muckler, Human Factors Review: 1984, The Human Factors Society, 167-208.

Williges, R.C., Williges, B.H., Elkerton, J. (1987) Software interface design; in, G. Salvendy, Human Factors Handbook; John Wiley & Sons, 141601449.

Willoughby, A.A. (1989), Making qualitative reasoning more quantitative , Avignon 89

Wood, S. (1988) Planning in a Rapidly Changing Environment , Report on 8th Workshop of the Planning Special Interest Group, Nottingham, IEE 1988.

Wright, M.L., Green, M.W., Feigl, G., Cross, P.F. (1986) An Expert System for Real-Time Control , IEEE Software March 1986.

Wu, W. (1978) On the Decision Problem and the Mechanisation of Theorem Proving in Elementary geometry , Sci.Sinica 21, 1978; also in (eds.) W.W. Bledsoe, D.W. Loveland (1984) Theorem Proving: After 25 years , Contemporary Mathematics 29.

Yeap, W.K. (1988) Towards a Computational Theory of Cognitive Maps , Artificial Intelligence, vol.34, no.3, April 1988

Young, R. M. (1987) Role of Intermediate Representations in Knowledge Elicitation. Research and Development in Expert Systems IV. pp. 287 - 288.

Zadeh, L.A. (1981) PRUF - a meaning representation language for natural language in (eds. E.H.Mamdani & B.R.Gaines) Fuzzy Reasoning and its Applications, Academic Press, 1981.

Zadeh, L. (1983) A theory of common sense knowledge. Technical Report Technical Report M83-26, University of California, Berkeley, April 1983.

SANDERLING

Final

April 1990

SANDERLING Final Report :
PART B Definition and Technology Analysis

Project Title :	SANDERLING
Project Number :	706.70040
Client :	ARE, Portsdown
Contract Number :	SLS21C/19
Authors :	Various from Logica, British Aerospace and Cambridge Consultants Ltd.
Status :	Final

Signed : S. Cogbill

Authorised : P G Jenkin

PART B : Definition and Technology Analysis

Table of Contents

1. INTRODUCTION

1.1 Background

Recognizing the potential to be gained in command and control (C^2) systems from knowledge based systems (KBS) and the related field of artificial intelligence (AI) the Admiralty Research Establishment (ARE) set up the Technology Demonstrator Programme (TDP) and the associated Technology Demonstrator System (TDS). This work is addressing many of the practical and immediate problems associated with the application of KBS to data fusion in the context of Naval C^2. The TDS project is progressing towards sea trials as an engineered data fusion demonstrator. The parallel TDP is providing the supporting research. In addition, ARE are undertaking other research via prototypes, theoretical studies and experiments aimed at establishing a wide capability in KBS based C^2.

Many of the functional characteristics and technical problems associated with Naval C^2 are similar to those in other large and complex military applications. In particular it was realised that there was significant commonality with the Strategic Defence Initiative (SDI)/Theatre Missile Defence (TMD) scenario. As a result a joint approach to research in the technology was agreed between ARE and the SDIO/SDIPO. A major part of that approach is a programme of research extending over four years. The programme will link to the current ARE research activities and will consist of a number of research projects to be undertaken by ARE, UK industry and universities.

The first phase is a 6 month study to define the contents of the research programme. The study commenced in October 1989 and is code-named SANDERLING. This report is part B of the final report for the SANDERLING study, and describes the process of technology analysis that took place in the first part of the study.

1.2 Aim

The aim of the SANDERLING study is to generate a KBS research programme in command and control for Naval and TMD applications.

1.3 Report

The results of the study are being reported in three parts. Parts A and B concentrate on the analysis phase of the project. Part A introduces the requirements of the study, the goals and objectives of the work and the approach and methods used. Part B shows how the six technology research streams for the programme were generated. This analysis is complemented by an applications perspective and culminates in a set of initial ideas for SANDERLING research projects which are described in Part C, together with a recommended research programme.

1.4 Part B

Following the completion of the definition phase, the next phase of the SANDERLING project was aimed at conducting a technology based analysis of the Naval and TMD applications. Part B of the final report describes this process to identify a list of technology based streams which could then be used as a basis for deriving a set of potential research projects. It also explores the research issues and priorities for sub-topics within each technology stream.

Section 2.1 describes the results of the brainstorming sessions held to identify the technology streams. The domain characteristics of the Naval and TMD domains were identified, and then technologies relevant to solving problems with these characteristics were derived. These technologies were then grouped together where possible into technology streams. Sections 2.2 to 2.7 describe the results of the analysis stage of the study which investigated each of these streams in more detail. The structure and scope of each stream is identified, primarily in terms of a set of sub-topics. Then the relevance to the goals and objectives of the Naval and TMD application areas is described. Finally, a set of prospective research issues and goals is given for each stream. The description of each sub-topic is supported by Annex B1 which contains a brief description of the state of the art, a list of the current activities in the U.K. and Europe, and a list of the available resources in the U.K. for each sub-topic.

Section 3 then describes an evaluation process which was applied to each of the technology streams. A set of research objectives and criteria are proposed, together with rules for applying the criteria to give a ranking of the importance of each sub-topic within a technology stream for each of the main objectives. No evaluation or ranking takes place across different technology streams. The results of the evaluation process for each technology stream is presented in tabular form, together with a brief interpretation of the results obtained.

2. TECHNOLOGY STREAMS

2.1 Definition of Technology Streams

The Technology Streams were generated from the enabling technologies that were required to support the Naval and TMD command and control applications. The first activity in the process of identifying the enabling technologies was to characterise the two command and control domains and identify the major functional elements. This was carried out in the first brainstorming session, and the results are summarised below.

2.1.1 Characteristics of Command and Control Domains

The characteristics of the Command and Control domains where identified and are given in the following table:

Naval	TMD
Distributed / Multi-agent systems	Distributed / Multi-agent systems
Sensors	Sensors
Weapons	Weapons
Organisational Architecture - existing	Organisational Architecture - new
Interactive adversarial mode of action	Set Plan mode of action
Un-cooperative agents	Un-cooperative agents
Variable tempo	High tempo
High and variable levels of Uncertainty	High and variable levels of Uncertainty
Temporal and Spatial problems	Temporal and Spatial problems
Layered Defence	Layered Defence
More Heterogenous Domain	More Homogenous Domain

2.1.2 Features of the Major Functional Elements

Both the Naval and TMD C^2 domains could be considered to consist of the following major functional elements. These are:

- Data Fusion
- Situation Assessment
- Resource Allocation
- Planning

Knowledge based system technologies have a potential contribution to all of these functional elements. A discussion of the domains, as well as the activities and potential sources for research within these major functional elements is given in Annex B1.

From the Brainstorm Meetings the major functional elements were characterised as having the following features:

2.1.2.1 Data Fusion

- Synthesis of possibly conflicting sensor data
- The design of suitable paradigms for data fusion
- Assessment of capability
- The integration of information sources with sensor data

2.1.2.2 Situation Assessment

- The analysis and representation of threats and engagements
- The analysis and representation of own resources
- The representation of weapons systems geometry and weapon states
- The ability to handle rules of engagement
- The need to take into account sensor coverage and weapon coverage
- Adherence to plan
- Representation paradigms which take into account the perception of each other by adversaries.
- The concept of defence screens

2.1.2.3 Resource allocation

- The navigation of the ship to avoid / engage enemy
- The deployment of sensors and the selection of the mode of active sensors
- The control of aircraft
- The deployment of decoys and ECM
- The assignment of weapons
- Adaptive preferential defence / firing doctrine
- Man-in-control vs man-in-the-loop

- The execution of these functions within the established organisation structure

2.1.2.4 Planning

- Adversary modelling
- Resource management
- Pre-hostilities planning
- Re-active planning
- Reconstitution and recovery planning

2.1.2.5 Shared Characteristics

- The need for high speed processing
- The need for risk reduction in the deployment of the technology
- The need for engineering standards
- The design of the man / KBS interaction, including :

    facilities for manual intervention

    interpretation / explanation

- The ability to handle multi-platform information
- Top-down issues of design and functionality
- The ability to exploit variable tempo
- The ability to deal with changing uncertainty :

    processing implications

    HCI implications

- Techniques for commissioning and tuning

2.1.3 Enabling Technologies

From consideration of the above list the enabling technologies were identified and investigated. Of particular concern was the synergy with other technologies and the relevance to one or more of the top-level goals of the programme which were defined as:

A Deployment of operational Naval data fusion systems.

B Enhanced functionality of the present systems.

C Application of the technology to the TMD domain.

The enabling technologies as intially considered were:

Technical Topic		Relevant to :	Synergy with :
1	Hardware Architectures	A,B,C	Real-time / Systems Neural nets Distributed AI
1.1	Paradigms		
1.2	Tools		
1.3	Architectures		
2.	Real-Time / Systems	A,C	Hardware Architectures Knowledge Representation
3.	Distributed AI	B,C	Real-Time / Systems
4.	Knowledge Representation & Manipulation		
4.1	Uncertainty	(A),B,C	
4.2	Spatial Reasoning	B,C	
4.3	Temporal / Modal Reasoning	B,(C)	
4.4	Deep Reasoning	B,(C)	
5.	Planning Technology	B,(C)	Knowledge representation Distributed AI
6.	HCI	(A),B,(C)	
6.1	Interface Paradigms	A,B,(C)	
6.2	Cognitive Ergonomics	B,(C)	
6.3	User Modelling	B,(C)	
7.	Development / V&V Methods	A,(B),(C)	
7.1	Life Cycle		
7.2	Knowledge Acquisition		
7.3	KBS Development tools		
7.4	Specification Methods		
7.5	Robust Architectures		
7.6	Maintenance		
7.7	Validation & Verification		
8.	Database/KBS interaction	A,B,(C)	
8.1	Coupling	A,B,(C)	
8.2	AI-enhanced Databases	B,(C)	
8.3	Intelligent Front ends / Nat. Lang.	B,(C)	
8.4	Novel Database Structures	B,(C)	
8.5	Dynamic Databases	A,B,(C)	
8.6	KBMS	B,(C)	

9. Related Technical Areas

9.1	Machine Learning	B	Methods
9.2	Neural Nets	B	Hardware Architectures
9.3	Situated Action	B	Knowledge Representation
9.4	Self-organising Fuzzy Logics	B	TMD Application
9.5	Pattern Recognition	B,C	TMD Application

10. TMD Application

Brackets denote that the technical topic is only partly relevant.

2.1.4 Research Streams

The full set of enabling technologies were analysed in the second brainstorming session, and consolidated into a final list of Research streams. This was used throughout the remainder of the study:

Technology Streams :

Research Stream	Incorporating
1. Hardware Architectures	1. Hardware Architectures 9.2 Neural Nets
2. Systems Design	2. Real Time / Systems 3. Distributed AI
3. Knowledge Representation and Manipulation	4. Knowledge Representation 5. Planning 9.3 Situated Action
4. HCI	6. HCI
5. Database / Knowledge Bases	8. Database / Knowledgebases
6. Methods	7. V&V Methods 9.1 Machine Learning

Application Streams :

7. TMD	10. TMD Applications 9.4 Self organising Fuzzy Logics 9.5 Pattern recognition
8. Naval Applications	

The two application streams were introduced to ensure there was sufficient focussing of the Technology streams on the application characteristics.

2.2 Hardware Architectures

This research stream is concerned with investigating those topics which are seen as critical in determining any system's eventual performance. By performance, we are primarily concerned with the speed at which the system is able to perform, and the number of computations it is carrying out within a specified time frame.

The stream's principle concern is not with hardware per se, but with means of exploiting hardware capability, such as through the use of parallel processing or application-specific hardware.

2.2.1 Structure and Scope

The research stream has been broken down to include four subtopics: Paradigms, Tools, Architectures and Neural Networks. A brief description of the role of each of these follows:

<u>Paradigms</u>

There are a range of paradigms available for implementing KBS solutions to a problem : production systems, semantic networks and blackboards for example. Each of these generic paradigms is usually tailored for a specific application. When considering new architectures for the solution to a problem (eg. incorporating the use of parallel processors) an analysis of paradigms for exploiting that architecture to the fullest should be done.

<u>Tools</u>

The use of tools to help produce, analyse and verify a particular solution to a problem is desirable, both in terms of generating a solution efficiently, and in ensuring that the chosen solution maximises a set of performance criteria. A range of tools already exists for the design and application of Knowledge Based Systems, and these should be expanded where necessary to assist the process of placing application code onto special hardware. Of particular interest will be tools which can determine the mapping of a particular algorithm or paradigm to an optimal processing architecture, and for measuring the subsequent performance on that architecture.

<u>Architectures</u>

The study of new software and hardware architectures for solving complex problems is a long-standing area of research. Attention is currently focussed on the potential that explicit parallelism has for the combination of numerous processors, each with their own local memory, but this should be considered as only one of a range of possible solutions, together with dedicated AI computers, associative memory type architectures, parallel processors working with a central memory. An appreciation of the problems of attempting parallel implementations of specific information processing algorithms is important. The recent progress made in these areas has led to the belief that a range problems will become tractable, both in terms of computational power required, and in their robustness to individual processor failure. There are also developments in other fields, such as storage media and increased communication bandwidths which have led to increases in computer performance and functionality.

Neural Networks

Neural networks are typically made up of a large number of simple processors, operating in parallel to produce associative memories, solutions for optimisation problems and classification systems. They have been the subject of a dramatic increase in interest in the last 3-4 years. They are usually "trained" on a set of known inputs and corresponding outputs, or just inputs (as in the case of unsupervised learning). The trained network is then used to process previously unseen inputs in an attempt to produce a valid output for that input. While neural networks cannot be thought of in the same way as the other subtopics mentioned here, they have been included in this section because of their obvious relationship with research into new processing architectures.

There is a high degree of interaction between these subtopics since none is independent of the others. For example, tools will likely support a particular paradigm on a particular processor architecture, which could be a neural net. The most direct interactions with other streams are with the systems engineering issues and the design of knowledge representation and manipulation systems that can exploit novel architectures.

2.2.2 Relevance to Goals and Objectives

Since each of the applications (Naval and TMD) is working in a highly time-critical domains, it is fundamental to the success of either that they achieve response times as required by operational requirements.

An investigation of different architectures and their effect on performance is of high importance to the Technology Demonstrator Programme, and will also be critical in designing systems capable of handling the computational load imposed by the TMD scenario. The success of the TDS will be critically dependent on the information processing algorithms, paradigms and architectures employed at all levels (Data Fusion, Situation Assessment, Resource Allocation, Planning). Expanding the amount of computing power available alone will not necessarily improve performance, especially when the scale of the problem increases as in the TMD scenario.

It should be noted that given the restricted timescale available, i.e. demonstration of experimental operational use by 1991, then research aimed at contributing to this must be well focused and capable of producing results within a short period, i.e less than 18 months.

2.2.3 Research Issues and Goals

A number of key technical issues raised in this stream will need to be considered when attempting to build future systems for C^2.

Speed : Meeting response times is critical in the military environment, and achieving a set of required processing rates are essential for all parts of the system. Knowledge Based Systems have tended to be developed without consideration for performance times. This is not surprising, since most KBS are developed using rapid prototyping, and the final behaviour cannot be determined from the outset. KBS also have the additional feature of often being discontinuous in their behaviour: the addition of a new rule can cause a dramatic change in performance. The problem can be tackled in several ways:

- Trying to formulate the problem in as detailed a manner as possible, and use this to choose the best architecture to implement its solution.
- Deciding on an architecture which has the potential for fast behaviour (e.g. parallel), and trying to adapt the problem to fit that architecture as well as possible.
- Taking an existing system, and look to improve its performance incrementally by importing a set of optimising techniques.

Functionality : Defining, and achieving the required functionality from a system is an important prelude to experimentation. Often, the functionality of a system is defined by its current status (i.e. what it does defines what it should do). Defining expected behaviour will be an important part of establishing criteria for judging the success of a range of experiments.

Robustness : Building robust systems is another fundamental part of the requirement for C^2. As well as the internal performance of KBS, this also applies to the overall architecture for an envisaged system. Will it be resilient to the loss of a sensor, or the loss of a computing unit, or the breakdown of communications ? Clearly, some situations will not be possible to overcome, but paradigms and architectures which distribute their processing and physical location can be more robust.

Communications : The provision of adequate communications is another essential requirement for the C^2 domain. New scenarios and architectures may place even higher demands on communications bandwidths than had previously been estimated.

There are two themes through which these issues could be explored in research,. The first is built primarily around the pragmatics of the programme as a whole, and involves research based on the rule and blackboard approach of the TDS systems. The second involves a revisitation of that approach and the consequent exploration of alternative paradigms, tools and architectures for tackling the fusion, situation assessment and overall control functions.

An overall goal of the research should be to utilise the new and expected developments in hardware in order to make significant steps towards developing a more non-pragmatic approach to the sub-problems of any C^2 system. If distributed and parallel approaches to a problem are seen to have benefits (either in terms of speed, or robustness), then these can now be researched actively due to the existence of new tools and hardware to support their development.

The sub-topics of tools and architectures are the subject of much commercial research already. The intention for this programme is to exploit as much of this as possible so as to avoid repetition. However, there are a range of tasks specific to the application area which would benefit from completely new tools, e.g. for system monitoring and performance measurement.

Novel processing mechanisms such as Neural Networks or Genetic Algorithms often display remarkable performance. It is important to build and maintain a balanced view of the contribution such new techniques can make to problem domains. The true test of any such technology lies in the demonstration of its applicability within the domain of interest. Since neural nets have to date been best applied to the efficient processing of large amounts of disorganised data for pattern recognition, they are a strong candidate for such exploratory work.

2.3 Systems Design

The concerns within this stream are those to do with architecture, rather than knowledge-based behaviour. As such the distribution of function across the many components of a command and control architecture, the robustness of the architecture, its communication needs and its dynamism are to be studied. Within each node of such an architecture, and across the nodes, the issues of speed of response (the consequences of interrupts and the prioritisation of processing), and memory management are also raised.

2.3.1 Structure and Scope

This stream has only two subtopics:

Real-Time AI Systems Design: This is concerned with several specific problems in the development of reasoning systems that have the capability to operate in real time. For example, these must deal with interrupts, be able to prioritise processing and guarantee response times. These are clearly capabilities humans have, and their abilities may form a model for computer implementation.

Distributed Artificial Intelligence: DAI is concerned with the design of co-operating knowledge based systems. It addresses issues such as distributing decision support in command and control structures that are already distributed in nature, and control and communication in such architectures.

The issue of the use of distributed processing, ie. parallelism, for increasing performance is deliberately taken to be a component of the Hardware Architectures research stream. The consideration here is with the natural mapping of function onto a distributed system - ie. where a function is already implemented by several people co-operatively. There is clearly going to be a strong interaction between the two streams because of this shared concern with parallelism.

There is also interaction with the Knowledge Representation and Manipulation stream since it has a concern with mechanisms for using knowledge and with efficiency.

2.3.2 Relevance to Goals and Objectives

Alongside considerations of appropriate hardware and software architectures for Naval and SDI command and control, attention has to be paid to the specific requirements that such systems should operate in real time. Care should be taken to distinguish such aspects from those relating to "real-world" applications where, for example, reasoning with uncertainty and incompleteness of data is important. Here, timeliness of reasoning is the goal, and the requirement is for systems that reason continuously about asynchronous events where a response is required within the timescale of those events.

Systems design is often influenced to a small degree by the application requirements and to a large one by the hardware and software architectures used. It is relevant to all functional levels within command and control, and to both SDI and Naval domains. However, its importance depends on performance goals placed on hardware and software.

In the DAI area we are concerned not with the exploitation of multiple processors for performance reasons, though that will be an outcome, but the consideration of problem solving as a co-operative process between several agents distributed across processors and space. This is an appropriate model in Command and Control domains because of the physically distributed nature of sensing systems, weapons and command systems. A review of previous exercise observations, for example, shows processed (fused) information arriving at a command centre from many points and planning being done through co-operation between a number of individuals. Much of the work in Distributed AI (DAI) has been inspired by such military applications as Data Fusion, Situation Assessment and Planning, as is reflected in the project descriptions in Annex C2.

2.3.3 Research Issues and Goals

<u>Real Time AI Systems Design</u>

In Real-Time AI Systems Design we can identify the following major issues.

Interrupts

We take the handling of interrupts to be a fundamental part of conventional real-time systems and it is no less important in real-time knowledge based systems. In a parallel architecture there are aspects of this problem related to the physical passage of the interrupt through the co-operating systems and how this is managed without bringing the entire system to a halt. But the most difficult issue is how to interrupt a "strand of reasoning" without jeopardising the consistency of knowledge and databases. A decision has to be made on whether to suspend this reasoning and resume it later or abort it and remove any trace of its work. Again this is difficult enough and poorly understood in single processor systems, and complicated by any move to parallel implementations. Even if it is assumed that all sensor input occurs on a polling sequence, the processing that is being performed in Data Fusion and Situation Assessment is inherently aimed at identifying particular threats. When this happens, the natural thing to do is to interrupt other processing rather than wait for the vehicle hypothesis with a high threat value to simply propagate through with others. Also, experience in the Air Defence domain has shown that there are a number of background processes being performed by planners that are immediately "shelved" when an event occurs in the evolving Recognised Air Picture. The objective, at the planning level at least, then, is to provide a mechanism to implement this process interruption.

Prioritisation

Associated with the processing of inputs is the ability of a system to reschedule its internal activities to deal with items of high priority, for example the detection of a suspected anti-ship missile. Mechanisms for focusing attention in single processor systems are difficult to design well, and to do so in a parallel environment places interesting constraints on the choice of appropriate co-operation strategies. Where insufficient computer power is available to process everything, focusing attention is crucial to functional adequacy.

If we make the assumption that sufficient processing power can be provided at the Data Fusion and Situation Assessment levels to process all sensor events, prioritisation still takes place at the planning and resource allocation levels. It is a fundamental component of reasoning in these resource limited domains that effort has to be expended in determining which are the most significant threats against which to deploy resources. The objective with regard to this issue is to understand how people prioritise their actions in these domains and replicate the ability within a knowledge based system.

The scarcer processor time is as a resource, the more judiciously it should be used. If there is faith that processor performance will meet all demands, then there is nothing to be gained, indeed time will be wasted, in managing processor utilisation.

Progressive and Deferred Reasoning

Another mechanism for guaranteeing response times is the use of a reasoning strategy that is guaranteed to return some answer, but the quality of that answer improves with the time available to produce it. This is related to prioritisation, the explicit consideration of time as a resource for reasoning, and its consequent management.

Reasoning in this manner is clearly within human capabilities, but the complexity of designing a reasoning system to replicate it is high. The question is whether a response must be guaranteed to any enquiry, if so there may be a need for this type of approach.

Garbage Collection

A practical consequence of large knowledge based systems, particularly those viewed as co-operating or multiple knowledge source systems, is the generation of garbage on a micro and macro scale. AI languages use dynamic storage allocation and need incremental garbage collection (as opposed to stop and sweep) if any kind of response time is to be guaranteed. Of far greater significance however is garbage management on a larger scale. In a co-operative and distributed system no one knowledge source has total responsibility for a piece of data that is shared, for example on a blackboard. Clearly in a parallel system such problems are exacerbated and require thoughtful design.

This problem is prevalent in command and control domains where there can be much spurious data and clutter from old track reports. There is no simple way of dealing with the problem either, since time-of-last-access to data is not necessarily a good indicator of its significance. Schemes requiring knowledge sources to periodically refresh their interest in pieces of data are time-consuming and difficult to use during system development and debugging. There is no simple objective that can be stated here, mainly a reminder that this issue is of pressing importance in systems design.

The sophistication required in Garbage Collection depends on the rate of memory turn-over. If objects are persistent, some kind of backing store will be required; if they are highly transient, memory will have to be reclaimed frequently. The higher the turn-over rate, the better garbage collection has to be.

Consistency

Another consequence of distributed data, and of dealing with real sensors, is the potential for different processors to be working with temporally separated data. There is then the possibility that a user or reasoning process looking at a global state will draw erroneous inferences. This may be more a fear than a practical problem since time differences of a minute or two will be insignificant when applied to aircraft or ship tracks. In the case of missiles, however, the problem may take on more significance.

A related issue is non-monotonicity. Since this is a more central concern in developing a reasoning strategy it is considered under the Knowledge Representation banner.

As we have stated before, these are largely engineering issues rather than technology areas so there is a great deal of domain dependence in any research work.

Other real-time issues

There are other considerations that constrain proposals for research on the issues of real-time systems design:

(1) The TDS programme already presents a stand point on these issues. The assumptions about performance of the rule-based paradigm on the chosen processors has eliminated the need for an interrupt driven scheme, for any prioritisation and for a time-constrained reasoning technique. There are garbage collection rules based on age of data and data type. Consistency is maintained by continually updating all hypotheses from new data arriving.

(2) The existing TDS architecture is already felt to have a performance that can be scaled up to the level required of a deployable system through either improvements in processor technology or a small degree of parallel processing. The necessity therefore to address the more complex issue of redefining the reasoning strategy to support, for example, progressive reasoning, is contingent on the requirement for marked performance improvement.

In view of the above a "wait and see" approach would be valid and appropriate. Both the TDS and TMD demonstrators will be subjected to bench marking exercises which will provide the evidence to justify investigation of the systems engineering issues discussed above.

Nevertheless, we know that the processing requirement in the TMD domain is going to be considerably greater than the Naval one, by at least two orders of magnitude. It is therefore sensible to look at performance improvements from better reasoning strategies as well as from parallel processing. Secondly, it seems certain that the rule-based formalism used throughout the Data Fusion demonstrators will be inappropriate for reactive resource allocation and planning. The issues associated with designing real-time reasoning systems within, for example, a frame or script paradigm are therefore worthy of investigation.

Distributed AI

In developing distributed AI applications there are a number of issues that need to be addressed:

- *Control* - centralised control offers tight and exact management but in any distributed system it can be a potential bottleneck limiting performance. Experiments have shown marked differences in performance in systems depending on the control regime under which they are operating (e.g. hierarchic or heterarchic).

- *Communication* - deciding what should be communicated and when it should be sent is non-trivial. There is an in-built assumption that communication bandwidths are limited and subject to potential data loss, but co-operation, and therefore the sharing of information and goals, is an essential part of the problem-solving process. No one node is taken to have the capability to solve the entire problem itself, though no other assumptions are necessarily made of the capabilities of each agent.

- *Partitioning Applications* - dynamic task allocation offers maximum flexibility, and some fault tolerance, but does not necessarily provide optimum performance. It is envisaged that some form of manual partitioning will be required for most applications to meet performance demands.

- *Consistency of views* - There can be serious difficulty in reaching firm conclusions when co-operating systems have inconsistent views. It is clearly desirable to minimise opportunities for inconsistencies at the design stage.

A lot of effort has been put into researching appropriate architectures for applying DAI to sensor interpretation problems akin to those ARE are handling in the Naval data fusion domain. There seems to have been little effort within the UK to either replicate this work or explore the issues independently as they apply to sensor interpretation for this type of domain (work has been done on robot sensor fusion).

A barrier to entry in to this as a research topic is the poor availability of tools such as CAGE (Nii et al 1988) within the UK. Similar tools are already being worked on in UK establishments (e.g. CADDIE, an IED project involving Logica and Essex Universities in the development of an experimental environment for multi-agent planning, and a distributed MUSE (IED project 4/1/1325)), but it will be 1-2 years before these are available.

With the high level of actvity in DAI research in the community at large, only very focussed work would be particularly productive.

2.4 Knowledge Representation and Manipulation

Knowledge representation is concerned with the encapsulation within a programme of knowledge germane to task performance. This knowledge can be of many kinds and will typically include domain-specific knowledge, knowledge of problem solving methods and common-sense knowledge. Representation languages are needed for each type of concept, fact, relation, implication, etc found in examining human performance in domain problem solving. Such languages must have expressive adequacy and computational tractability.

2.4.1 Structure and Scope

The broad concerns in Knowledge Representation research are the development of languages with expressive adequacy, economy and efficiency of reasoning. Representations form a part of two elements within a system's design: the knowledge base and the database used during reasoning, ie. there is a need to describe what it is we know about the domain and about the current situation in appropriate languages. These two categories are quite distinct, there are criteria important to the latter that have far less significance to the former, and vice versa, the principles of explicit naming and least commitment for example.

Many issues arise concerning appropriate forms for representations, eg. procedural, declarative, qualitative, probabilistic, analogical, which can only be resolved with reference to a domain's characteristics. From our earlier consideration of the nature of Naval and TMD command and control (Section 2.1), we have identified the areas of Uncertain, Temporal, Spatial and Deep reasoning as being particularly relevant, along with mechanisms for non-monotonic reasoning and multiple worlds, planning and Situated Action. We briefly consider each of these areas below.

Uncertain Reasoning

This can be considered a rather fundamental aspect of expert reasoning, and as such it has been the subject of research for many years. In decision making, the weighing up of different options is seldom a "cut and dry" affair; various alternatives will have their pros and cons, and sometimes delicate judgements need to be made about their relative value.

Spatial Reasoning

Points, Curves/Lines, Planes, Regions, Angles/Bearings, Volumes/Shapes, are spatial components of the reasoning in some domains. Groupings, Relations and Distributions of objects are also examples of spatial characteristics. These will typically form an important element of the reasoning in domains that have a visual component, either with an operator looking at isomorphic situation displays or at the scene itself.

Temporal Reasoning

This is not concerned with systems that reason in real-time but with reasoning explicitly about time and temporal relationships between actions and events. People can reason about the past, the future and the present, they can divide time into smaller and smaller intervals, but the resolution used is dependent on the specific task in hand.

Deep Reasoning

"Deep" is an upsetting term for many workers in the KBS field because it lacks, and evades, rigorous definition. A reasonable definition would be that a system uses Deep Representations if it has explicit descriptions of concepts that in other implementations would be implicit within other knowledge bases. If we take the Production Rule approach of the Data Fusion demonstrator as the baseline, then explicit spatial and temporal representations would be considered deep. A further relevant dimension is added by methods for modelling intentions and beliefs of other agents, ie. the potential to model adversaries' and other forces' behaviour.

Non-Monotonic and Multiple Worlds Reasoning

In systems doing continuous hypothetical reasoning on asynchronous data the potential for newly received data to supercede previous input, and reasoning performed on it, is high. Managing this process is the subject of Reason Maintenance Systems.

Related to this is the idea of reasoning about the future, generating several alternative hypotheses (worlds). This is useful either when the future cannot be predicted and no commitment is wanted until the actual situation becomes clear, or when there are many alternative actions and each needs to be explored, eg. in planning.

Planning

While many things are often termed "planning", the general view is that planning is the choice of an arrangement of actions that achieves a set of goals under constraints on time and resources. For a real-time, or just a real-world, system, the challenge is to plan while goals, possible actions, timescales and resources may change outside the influence of the planner, and when the actions actually performed may not conform to the plan, or achieve what was expected of them.

Situated Action

The term "Situated Action" refers to a theory concerning human goal-oriented behaviour; it contrasts plans as a description for such activity with actual behaviour. It suggests that actions are *contingent* (i.e. dependent on circumstances found during actions) and *emergent* (i.e. will develop from circumstances over time). Plans are a resource for action - they do not fully describe actions. Expertise can lie in the ability to keep generating new avenues and exploit serendipity.

2.4.2 Relevance to Goals and Objectives

Our investigation of the Naval domain, in particular a set of reports on exercise observations produced by Electronic Facilities Design Ltd (EFD 1986, 1998a, 1988b, 1989a, 1989b), have shown the practical relevance of the topics we discuss in the following sections. These reports relate, and in many cases provide transcripts for, specific instances of command teams using temporal, spatial and epistemic reasoning in their situation assessment and planning activities. They also show command teams considering the quality of information they are receiving, and sending to other people, their revision of hypotheses (and hypothesis fixation) and use of what we would term "multiple worlds" hypothetical reasoning. We can therefore feel highly confident in our justification for research in these areas.

The principle relevance of this stream is to enhanced functionality of the demonstrators in both Naval and TMD areas. Each sub-topic has different characteristics as discussed below.

Uncertainty

Uncertainty stems from a number of factors: the domain knowledge being applied, the sensing devices being used and deliberate deception being particularly relevant within the Naval and TMD contexts.

A related issue is incompleteness of data. Since sensor coverage is restricted and may be jammed, only parts of the complete scene will have data being reported from them. Similarly there are gaps in knowledge, particularly about what emitters may be doing and what tactics may be used by the enemy.

For a thorough review of the origins of uncertainty in the Naval domain see (Moon et al 1988).

Of crucial importance in Command and Control applications is the risk associated with decisions that have to be made. Using uncertainty in decision making must involve an assessment of the cost of making a wrong decision; it is better on the whole to have played safe than taken the most certain line. This attitude to decision making can have extensive system effects since it may involve the generation of contingency plans, with all the reasoning needed to support them.

In TMD not only is reasoning based on very low quality knowledge and information but the consequences of error are grave. Under these circumstances some very domain-specific reasoning mechanisms are probably justified.

Spatial Reasoning

There are many examples of spatial entities being reasoned about in Naval and TMD domains. Weapon arcs, Jamming Spokes, Exclusion Zones and Defensive Screens are common examples in the Naval case : distributions of objects in threat tubes can be informative in the TMD problem. Groupings of tracks, particularly of hostiles is important in the Naval domain for looking at force structures and deciding on their likely intentions.

There is a collective significance to the tracks and spatial features of both domains. The scene understanding done by commanders looking at situation pictures is substantial, scenarios are largely described and understood in pictorial form. Explicit quantitative and qualitative symbolic descriptions are therefore needed in picture representation, along with inference mechanisms for use in Situation Assessment and Planning. These higher levels of processing must "see the picture" rather than a list of tracks.

Temporal Reasoning

The temporal evolution of scenarios is an intrinsic part of the domain. Naval operations take place over periods of up to several days, if not weeks. During that time resources will be scheduled to do things at specific times, to wait for things to happen, to look for sequences of events.

Overt reasoning about time is essential at the planning level of command and control. The planner is establishing actions for a collection of resources which may well be contingent on the way the scenario evolves. The actions of each resource in the plan will be interlocked and dependent not on an absolute time, but the time at which a previous action is complete, or has started. Representing and reasoning about such a plan has to involve a temporal component.

Temporal reasoning has less relevance to the TMD domain. There are instances where time is an important consideration, such as planning waves of weapon releases, but in general these should be within the capability of ad-hoc methods.

Deep Reasoning

Limiting this to adversary modelling relegates its role to the Situation Assessment and Planning levels. In Situation Assessment predictions of likely intentions are a component of threat analysis, and in planning are a component of the "Game-Theory" approach to force deployment.

As there is a degree of autonomy in Blue Force platforms' actions there can be a requirement to model own force behaviours as well as the adversary's.

Again, in the TMD area the relevance is lower because there is more predictability of where objects are going to go. There is little scope for changing anything once the missiles are in flight.

Non-Monotonic and Multiple Worlds Reasoning

Through Data Fusion and Situation Assessment, inferences on new data can be allowed to propagate continuously to produce an up-to-date picture without using a non-monotonic reasoning system. When a decision had been made that cannot be easily or continuously modified, ie. in Resource Allocation or Planning, it becomes necessary to have some mechanism for detecting those changes in situation that affect decisions. Reason Maintenance systems are designed to do this in a general way, domain specific mechanisms can and have been devised for doing the same thing in an ad-hoc way. If significant events (ie. those likely to affect decisions) can be pre-defined they can be trapped and used to trigger replanning.

In plan generation in command and control it is clear that multiple hypothetical evolutions of the scenario are considered. Plans are generated in a robust fashion, geared for the most likely developments but considerate of the fact that the scenario may not develop that way.

Planning

The ARE view of planning in the Naval context is that it is composed of long- and short-term elements, with a nomenclature of Planning and Resource Allocation respectively. This view is certainly reflected in Naval practice with a clear separation existing between the functions and responsibilities for each. This sub-topic covers both aspects.

In Theatre Missile Defence, planning can be thought of in much the same manner if the domain is viewed as Extended Air Defence. Anti-Ballistic Missile defence per se is dominated by the resource allocation issues of weapon assignment. Longer-term (or larger-scale) issues such as preferential defence and firing doctrine would fall into the "planning" side of the dichotomy.

Situated Action

It is clear from exercise material that plans do exist for forces and that the plans are being continuously updated in repsonse to the developing scenario. The principles of Situated Action are typically of use in analysing and "debugging" peoples' behaviour. Since this will not be a prominent role for KBS in command and control it is not necessary to consider this sub-topic further.

2.4.3 Research Issues

All of the sub-topics but "planning" deal with the development of appropriate representation languages and thus have a common set of issues that we describe in the following sections. Planning issues are dealt with separately later.

When we wish to represent the world, when we want to model the changes and processes that might occur in it, we require an expressive medium - a knowledge representation language (KRL). In the following sections we highlight important characteristics that a KRL must possess. We then concentrate on perhaps the most powerful class of KRL - logic based representations

2.4.3.1 Syntax and Inference in a KRL

Any knowledge representation language has a syntactic and an inferential aspect. The syntactic aspect concerns the way in which one stores information in an explicit format. The inferential aspect concerns the way in which the explicitly stored information can be used to derive information that is implicit in it.

A knowledge representation language is a formal language. The first task when one is defining a knowledge representation language is to specify precisely what expressions are part of the language and how these expressions can be combined to construct new expressions in the language. A user needs to be told exactly what sets of symbols count as well-formed expressions in the language in order to be able to use the knowledge representation formalism to represent information about the world.

Apart from a syntax, each knowledge representation language also has an inferential aspect. A knowledge base will always represent explicitly certain pieces of information. However, by reasoning one might be able to derive information that is implicit in the knowledge base. Each knowledge representation language has its own preferred way of drawing inferences.

It is important to distinguish these two aspects of a knowledge representation language. There are usually two types of argument advanced for preferring one knowledge representation language over another. The first type relies on the syntax. Such arguments usually stress the naturalness and the expressiveness of one knowledge representation language over another. The second type of argument refers to inferential aspects. Arguments of this type usually draw attention to the power of the underlying inference machinery.

2.4.3.2 Levels and Knowledge Representation

There are at least four different levels at which one can talk about knowledge representation formalisms.

The first level is the implementational level. Knowledge representation languages are intended to be used to represent information on a computer, and it is therefore essential that it be possible to build a computer program to implement the knowledge representation language. This is the main concern at the implementational level.

The second level is the logical level. The main concern at the logical level is the logical properties of the knowledge representation language. There are two main questions that are relevant here. First, there is the question about the meanings of expressions in the formalism. One needs to know what types of information can be represented in the formalism. Second, there is the question about the expressive power of the formalism. An example of a question that arises here is the soundness of the inference procedure. An inference procedure is sound if whenever the input expressions are true, then the output expressions are true as well.

The third level at which one can discuss a knowledge representation formalism is the epistemological level. At this level one is concerned with discovering the types of primitive that are needed for representing particular pieces of knowledge without considering which particular primitives are needed.

The final level is the conceptual level. Whereas the epistemological level is concerned with the types of knowledge structuring primitives that are needed, the conceptual level concerns itself with the actual primitives that should be included in a knowledge representation language.

It is important to keep these four different levels distinct. A lot of the discussion in the literature is rather confused because knowledge representation formalisms that are preferred at one level are often attacked because they are not adequate at another level. The attendant arguments often fail to keep track of the level at which they are germane.

2.4.3.3 What do we want from a KRL?

There are various criteria which we can use to assess the value of a knowledge representation language. In this section we will discuss some of them. Each of the four levels at which knowledge representation formalisms can be analysed has its own criteria of adequacy, and a completely satisfactory knowledge representation language would of course meet all these criteria.

At the implementational level, the main criterion of adequacy concerns efficiency. An implementationally adequate knowledge representation language should allow one to store information in a space-efficient way, i.e. without taking up too much of the computer's memory. It should also draw its inferences in a time-efficient way, i.e. it should draw its inferences as quickly as possible.

At the logical level, one is concerned with the logical properties of the knowledge representation language. A logically adequate knowledge representation language should ideally have a clear semantics. It should be clear about the meanings of the syntactically well-formed expressions. Moreover, its inference rules should be sound. If the information that is explicitly stored in the knowledge base is true, then the implicit information that can be retrieved using the inference rules should also be true.

At the epistemological level, the relevant criteria of adequacy have to do with the naturalness with which representations can be constructed and understood. There are various epistemological criteria of adequacy. For example, if there is a natural way to organise information about a particular domain, then an epistemologically adequate knowledge representation language should be able to reflect this organisation.

Secondly an epistemologically adequate knowledge representation language should be modular so that whenever a particular piece of information changes, only a small part of the knowledge base would have to change.

A third epistemological criterion of adequacy is closely related to the previous two. It concerns the granularity of the knowledge representation language. Knowledge has to be stored in chunks. The granularity of the knowledge representation language determines the size of the chunks in which the knowledge is organised. Clearly the domain of application should not contain important information that it is beyond the 'grain size' of the KRL to represent.

A final epistemological criterion of adequacy concerns the relation with the conceptual level. An epistemologically adequate knowledge representation language should support whatever actual primitives one chooses at the conceptual level. Suppose that we decided that the best way to represent knowledge was as a set of facts, where each fact was represented as a predicate followed by a set of arguments. Then an epistemologically adequate representation language would allow the user to choose whatever predicates they liked at the conceptual level.

Conceptual criteria of adequacy concern the conciseness with which particular pieces of knowledge can be represented. If it is impossible to represent a particular simple piece of knowledge in a concise way, then the actual primitives that are used at the conceptual level must be wrong, and need to be changed. Similarly, if a particularly simple inference can only be made in a very complicated way, then one suspects that the actual inference procedure used is also inappropriate.

2.4.3.4 The Special Status of Logic as a KRL

Much of what appears in the literature about representation and reasoning is cast in terms of logical systems. It is worthwhile discussing why logics should be preferred.

There is a lot of confusion in AI about what exactly a logic is (see e.g. Israel and Brachman (1981) for a catalogue of confusions). It is therefore necessary to define the term logic precisely.

Logic can be defined as the study of correct inference. Although there might be some disagreement about exactly what makes an inference correct, there can be little doubt that minimally it should be impossible for the assumptions on which the argument rests, the premises, to be true while the conclusion is false. Minimally, a correct inference is truth preserving: if the premises are true, then the conclusion must be true as well. There might be many other conditions that have to be met, but logicians have taken truth preservation as their only criterion. Thus, logic is the study of truth preserving inference.

2.4.3.5 Issues in Planning

Real-time planning presents some interesting problems, not least because the system must try to make as much use as it can of existing plans, which may actually be being executed while the system is replanning. To do this, a planning system must be able to introspect on its own reasoning to determine problems in its existing plans (i.e. detect bugs) and then decide how best to remove them or nullify their effects.

The following sub-sections examine issues for representation and reasoning in each of the constituent parts of a plan. This list is itself incomplete, but there are further issues still that arise if there is user interaction during the planning and replanning process, particularly if the user is able to construct partial plans himself, set constraints and reconstruct plans.

In the view of planning as the arrangement of actions that achieves a set of goals under constraints on time and resources, issues arise associated with goals, actions, constraints and plans:

Goals

Goals usually manifest themselves as one (or more) of three forms: *What* must be brought about, *When* must it have been achieved, and *Where*. But there are many complications heaped on top of these simple characteristics:

Can many independent goals be represented?

In most situations humans find themselves in there is seldom just one goal that has to be satisfied. Choosing a path that goes some way to meeting many goals, or not conflicting too much, is the usual requirement. There can be many types of goal, from ones that are the principal aim of the planner to those that have to be maintained during all activities, e.g. ensuring fuel levels using air-to-air refuelling. What sorts of goals are present in the command and control application, how they are related and how they can be integrated need to be considered.

How are subgoals, goal conflict and support represented?

The goals that the planner is presented with will have various relationships. The simplest of these is the subgoal, where its achievement goes part way towards satisfying another goal. Goals can conflict with and support each other in many ways, either because of common resource usage or more fundamentally, e.g. high payoff and low risk, performed quickly and reliably. In such cases there is a degree of compromise that needs to be reached. There can be temporal relationships between goals, i.e. one thing being achieved before another, or in parallel.

Can all goals be predefined in objective and time?

If this is the case then the planning task is considerably simplified. If goals cannot change or be added to, then a solution can be developed and only modified if its execution fails; alternative solutions can already have been proposed in the event of failure. If goals change over time, then the current set has to be determined from time to time and replanning initiated which may substantially alter the executing plan. If goals are going to change then perhaps it is not necessary to completely plan a solution to all goals beforehand, and planning could take place continuously as execution proceeds. Since this is unlikely to lead to an optimal plan, it may not be an acceptable route.

How critical is any goal?

Along with representing the goal itself, there are likely to be several factors that need to be considered in relation to the goal. For example how critical is the goal and does that vary, is there merit in partially satisfying a goal?

Actions

Original work in planning was based on a backward chaining method that took a goal, searched for actions that could achieve that goal, then took the conditions that the action needed before it could be performed and made those into new goals. In such systems the action was therefore expressed as a set of preconditions and postconditions. This scheme has had several embellishments in later systems. In real life things are seldom so simple and many questions arise:

Must pre-conditions hold throughout the performance of the action?
How can precondition variables be externally affected?
How is action performance to be achieved?
How is action performance to be monitored?
Can pre- and post-conditions be directly measured?
What certainty is there that the action will achieve a result?
Does that certainty vary with conditions or time?
What other things will be achieved as side-effects of the action?
Are there alternative actions available for any goal?
How are choices made between alternatives?
How is the resource requirement of an action expressed?
Can actions be partly performed?
What does a partly-performed action achieve or change?
How should the duration of an action be expressed?

Constraints

There is a fine dividing line between some types of goal and what are more easily thought of as constraints. In a sense, a goal must require some action for it to be achieved, whereas constraints restrict the choice of action. Avoiding danger is therefore a constraint when threats are known a-priori, but takes the form of a goal when threats appear during a scenario.

The management of resource usage forms a major part of the consideration of constraints, since each action will usually require some resource for its execution, be it a sensor, a missile or ship, fuel or chaff. Issues that arise would therefore include:

What can each resource be used for?
Are resources consumed and can they be replenished?
Can resources be used on parallel actions?
Do some resources have to be used in combination?
Is there a cost associated with using a resource (e.g. active radar)?
How is the availability of a resource determined?

How these characteristics are made available for use by the planning system needs to be addressed in choosing representations for resources.

The other principal constraint the planner works with is time. The performance of any action is going to take time, it will vary from action to action and may vary for a particular action depending on the context in which it is performed. Goals will need to be achieved at certain times, which may again vary with circumstance. All of this, too, needs to be considered by the planner and incorporated into its representations.

Plans

How are goals, actions, dependencies and constraints stored together?

(To represent the plan). To introspect in replanning, more than just the end result, i.e. the action sequence, needs to be stored. This leads to a very large and interlinked data structure.

How are alternatives or contingencies held?

Alternatives being other ways of achieving the same goal under the same conditions, contingencies being things to do if the first thing you tried went wrong or didn't work. Under what conditions would these be used?

Is there a measure of quality that can be associated with the plan?

And, if so, is a measure of optimality useful to the planning process?

Is there a measure of robustness that can be associated with the plan?

I.e. what is the likelihood of this plan actually being achievable, and can that be used to shape decisions?

How are reasons for choices associated with parts of the plan?

Aah explanation! Having built a considerably interconnected structure of plans, actions, etc., whereabouts are the reasons for any choice recorded?

How is a partially backtrackable structure of plan derivation recorded?

In replanning part of the overall plan structure, how much of the previous reasoning can be retained, and in what form is it kept?

2.4.4 Research Goals

For the Knowledge Reepresentation and Manipulation research stream as a whole the goal is the principled study of these matters within the two domains. By this we refer to specific identification of objectives for a knowledge representation language and inference mechanism with respect to characteristics of human abilities at the tasks being supported or automated. If the representations to be used in building systems are to be robust, or compatible between several modules or support incremental development they must be given more attention than they have so far attracted.

It seems tautological to declare that knowledge representation is **the** key to building knowledge based systems, but the fact is seldom recognised as such.

2.5 Human Computer Interaction

This research stream is concerned with investigating a number of aspects of the human-computer interface. Research into HCI is multi-disciplinary and hence it is difficult to produce a categorization of research topics which does not appear somewhat arbitrary. The objectives of research are essentially to design better interfaces and to improve in as broad a manner as possible human interaction with computers.

2.5.1 Structure and Scope

Human-computer interaction, or the human-computer interface is rather an amorphous subject, analagous to psychology, straddling many scientific disciplines. Indeed, the three UK Research Councils (ESRC, MRC, SERC) in recognition of this situation and the potential problems that arise from it, recently established a Joint Council Initiative in Cognitive Science/Human-Computer Interaction.

The research stream for this study was divided into subtopics as follows:

Physical Interface: This embraces the controls, displays and 'dialogue' which an operator uses to interact with the computer. Research concerns the physical and perceptual aspects of the design of these system elements.

Design Methods and Tools: This refers to the various analyses, such as task analysis, goal analysis, knowledge elicitation, allocation-of-functions, etc. for specifying the users' requirements, and to the various techniques employed to carry out these analyses. The methods and tools span the entire system development process and hence also include prototyping and evaluation activities.

Modelling Issues: This covers how users form models of their world (domain models), models of the system and problems of system 'transparency', and how to construct models of the user within the system (i.e. embedded user models or adaptive interfaces).

Cognitive Issues: This is concerned with various aspects of the user's cognitive abilities and limitations such as, for example, reasoning with uncertainty, hypothesis fixation, confidence. It concerns explanations (i.e. computer explanations to the user) and includes the familiar problem of mental workload.

User Support: Referring to issues such as training (both on-line and off-line), help facilities, and the provision to the user of decision aids such as predictive displays.

2.5.2 Relevance to Goals and Objectives

The major problems which confront users of computer-based systems are not those of the physical machine (which, of course, remain important), but instead concern the manipulation or management of information. The 'human' in HCI is primarily interacting with information, with program logic, with knowledge or another 'intelligence'. Although this interaction takes place through a computer and its peripheral devices it should not be allowed to obscure the fact that the interaction is essentially cognitive and that the most important issues are cognitive. As stated in the original proposal, the form in which the information is presented and the way action is initiated must involve the user in a "seamless decision making process". The design of the HC interface is therefore highly relevant to the goals and objectives of the research programme. This is true of TMD as well as Naval application areas. Although a TMD system might be totally automated, it is very probable that a human operator will remain "in the loop" at some high level in a monitoring or supervisory capacity.

The human-computer interface is the medium through which the operators interact with the system. It is self-evident, therefore, that in order to fulfil objectives such as better evaluation of threats, avoiding hypothesis fixation, maintaining plan integrity and so forth, then improving the HC interface is essential.

The existing programme at ARE has constructed a display for the TDS with a view to operational deployment. The HCI stream is therefore likely to contribute more towards enhanced functionality and TMD application.

2.5.3 Research Issues and Goals

We have considered the technical issues (or technical barriers to be overcome) for each of the sub-topics in turn. With regard to the physical interface we have concluded that it is, broadly speaking, a mature technology. Although the computing technology in the military sphere lags behind that available commercially, and consequently state-of-the-art displays and windowing facilities cannot be taken for granted as available,the design of HCI controls and displays is largely not a problem. What *is* a problem is the design of the interactive dialogue and the presentation of information to enable users to carry out their tasks effectively. Some of the HCI design problems inherent in current shipboard C^2 systems were listed by Osga (1989):

- Terminology is mis-matched between functions, displays and manuals.
- Terms are vague and inconsistently used.
- Error messages are uninformative.
- Errors force task restart and may render system inoperative.
- Alerts are too numerous.
- Displays are "data-dumps" and not task supportive.
- Procedures force numerous shifts between displays and controls.

- Users must memorize numerous (up to 12) procedure outcomes.
- Visual feedback during task performance is non-existent.
- Tactical displays are "cluttered" and dense.
- No help information on-line.
- No database query facilities, burden of information extraction all on user.
- Displays use primarily text with no graphics.
- Related information is divided among numerous small displays.
- Information is not integrated to suport critical decisions.

The key technical issues in the physical interface area are therefore how to integrate displayed information; how to improve the transfer of information; how to use graphics appropriately.

It should be noted that we have not considered here recent developments of multi-media interfaces, virtual interfaces, Helmet Mounted Displays (HMDs), and gestural input devices. There are undoubtedly technical issues of concern about them, but in our view they are peripheral to the technical issues facing the design of the "conventional" HCI.

With regard to design methods and tools, there is a large body of HCI knowledge available to the system designer in the form of HCI guidelines, standards, and principles (eg HUSAT, 1988; Smith and Mosier, 1986; Williges et al, 1987), but a complete and satisfactory HCI design methodology has yet to be produced. Indeed, the subject of how to incorporate human factors HCI design principles into structured analysis and design methods (SADMs) is the subject of current research (eg Anderson, 1988). On the subject of design tools, it is evident that they have proved useful since they reduce programming effort and allow the rapid prototyping of interfaces which may then be criticised by the end-users more easily than paper specifications. The problem, however, with most, if not all, these tools is that although they enable faster production of interfaces the quality of the interfaces is not necessarily better. That is, it is just as easy to build bad interfaces with the tools as it is to build good ones.

The key technical issues in the design methods and tools area are therefore how to design interfaces that are appropriate for the tasks the user wants to carry out, that is, based on his task model; how tools can address the overall design of human-computer interfaces, of dialogue design, and of task analysis; how use of such tools including UIMS might fit into any software design methodology.

With regard to modelling issues, many if not all the models in existence are very cumbersome to apply because they require the detailed consideration of interfaces for which the design has been stated explicitly; this may not be possible or practicable in early design stages. With the exception of the CLG (Command Language Grammar) technique, the models are all techniques of evaluating user interface designs and not design generation. Even CLG has been criticised for being really a description of a designed system and for its use at an early stage having not being demonstrated. It is the case that while these modelling techniques (ie GOMS, TAKD, TAG, etc) do address aspects of the HCI, they cannot be used collectively and none address the domain.

The key technical issues in the modelling issues area are therefore how to improve the descriptive and predictive power of the models; how to improve the usefulness of modelling to HCI design and increase its "attractiveness" to interface designers who do not currently develop such models; how to fit the modelling techniques into the system design process; to determine if adaptive interfaces have useful applications and to understand the full implications for the user of such interfaces.

With regard to cognitive issues the general comment can be made regarding the various theories of information processing, problem solving, decision making and reasoning that their relevance to "real-world" situations like Naval C^2 systems is largely unknown. Indeed, the study of real-life aspects of cognition has become a major topic of research in itself (eg Shlechter and Togila, 1986). Furthermore, on the subject of mental workload it is true to say that while our understanding of workload has undoubtedly increased over the years, it has not increased in proportion to the number of publications. The AGARD (1987) review observed that little progress had been made since the previous review ten years earlier, and concluded that it was unlikely that any significant improvement in the state-of-the-art would occur during the next decade. It would appear therefore that workload research has reached a state of diminishing returns.

The key technical issues in the area of cognitive issues are therefore to determine the relevance or applicability of research into problem solving and reasoning to the design of C^2 systems; to determine if human biases in reasoning occur in such contexts and how to avoid them; how to design the user interface to ensure attention to relevant information and prompting for forgetting of earlier decisions; how to design better explanation facilities to KBS-based systems; to increase the generality of experimental results on workload; how to improve techniques of workload prediction; to understand what design features make some systems impose unacceptable levels of workload; how to utilize operational data for the assessment of workload.

With regard to organizational issues, for any multi-man system to function effectively there must be effective partitioning of tasks between the team members. In order to carry out task allocation a number of aspects need to be addressed. These include:

1. Authority and responsibility.

The mechanisms for allocation of authority and responsibility are largely governed by command style. In C^2 applications the usual command style is consultative, that is, decision making is vested in a single individual acting in consultation with subordinate specialists. As the tempo of events increases, however,the style may become more autocratic.

2. Group Dynamics.

The mechanism for sharing tasks between individuals will depend on the degree to which their relationship is collaborative or co-operative. In a *collaborative* relationship individuals share responsibility for task completion. The benefit of this relationship is that it allows true parallel processing of events. In a *co-operative* relationship each individual is responsible for specific tasks. In this case parellel processing is dependent on task partitioning imposed in a fixed manner or under the control of a supervisor. Although the work capacity is much higher with a collaborative relationship there are significant costs in terms of training and provision of facilities.

3. Group structure.

The normal structure to be found in C^2 applications is hierarchical with significant links across the hierarchy. In Naval applications there are some complexities in the hierarchy due to differing goals between own-ship and Task Group warfare command chains.

The key technical issues in this area are therfore what is the impact of the above organizational issues on the provision of facilities for Naval C^2 systems; to determine what effect KBS-based systems will have on the higher levels of organization; should team structure be changed to accommodate changes in information from a KBS-based system.

Although some sub-topics will emerge as having more relevance than others, all topics require some attention in the design, implementation and validation of a viable HCI.

2.6 Database / Knowledge Base Interaction

Provision of database facilities for Knowledge-Based Systems (KBS) is necessary to provide secure and efficient storage of large quantities of information to a variety of inferencing processes possibly running on different platforms. There are a number of strategies for doing this that can be envisaged : coupling KBS tools and databases; adding KBS functionality to database systems; adding database-type storage mechanisms to KBS tools; or a new class of software tools that combines the strengths of both. Exploring these possibilities forms the main part of this research stream. In addition, effective support for data fusion may include components that lie outside the class of presently-defined KBS techniques. Consideration will therefore be given to novel database architectures (such as Object-Oriented Database) which may be of assistance in this area. As well as the architectural considerations, the type of support given to data access by both applications and end-users in terms of intelligent data access will be explored. The significant contribution here is seen as being in the servicing of higher-level queries and of the efficient generation of appropriate data paths.

2.6.1 Structure and Scope

There are six sub-topics in this stream. These are listed below.

<u>Coupling</u>

One of options in considering database and KBS integration stresses the different, but complementary nature of the technologies, and does not seek to transform either, but merely to allow them to talk to each other. Technical solutions may vary from loosely coupled architectures to ones in which the KBS component takes over responsibility for a large part of the data transfer task. The implications of these different architectures will be looked at, with specific reference to the performance characteristics and timing issues.

<u>AI-enhanced databases</u>

A number of database vendors are, not unnaturally, pursuing this route of adding KBS functionality to database systems. There must clearly be some compromise of the flexibility of the KBS support offered by this route. Issues that will be addressed in this section include the effectiveness of extensions to data query languages, the extent to which this architecture can support dynamic data structures and currently being-researched problem areas.

<u>Intelligent Front-Ends / Natural Language</u>

This section will address the possibilities for supporting higher-level database queries through the use of intelligent front-ends. This intelligence takes the form of application knowledge that can be used to ensure only valid requests are actioned, but also data-access related intelligence to ensure that the most effective data paths are used. The ability of these to be couched in natural language will also be explored, particularly the measures necessary to ensure complete dis-ambiguation of requests.

<u>Novel Database Structures</u>

This section will examine work currently being done on novel database structures. An example of this is Object-Oriented Databases which enable the persistent storage of the types of flexible data structure previously restricted to in-memory manipulation. Questions marks currently exist as to the ability of these databases to handle the amount of data necessary for Data Fusion or TMD. The benefits of the technology, however, in terms of providing support for 'triggers', 'methods' or 'procedural attachments' for data-driven applications require evaluation. Other forms of database that can support real-time operation need consideration with respect to the demands likely to be put upon them by inferencing.

Dynamic Databases

The ability to maintain and make accessible real-time data from a profusion of sources makes demands on databases that are not catered for by commercial Database Management Systems. This section will explore the nature of these demands with special reference to the KBS inferencing cycle. A range of techniques may need to be examined to provide a database design with the necessary performance characteristics and integrity mechanisms, but will include strategies for managing the commit process. The implementation of a two-phase commit without read-locking is an example of such a strategy.

Knowledge Base Management Systems

This architectural solution involves bringing the KBS and database component closer together as a Knowledge Base Management System. Since the system controls the inferencing as well as the database access a number of performance advantages can be gained by the implementation of different memory management techniques. This section will concentrate on the effectiveness of currently available solutions as well as investigating the medium-term potential. This reflects the relative immaturity of work in this area.

2.6.2 Relevance to Goals and Objectives

The question arises as to whether it is relevant to investigate the combination of expert systems and database technology as part of the research programme. In considering this question, firstly, it should be asked what an advanced C^2 system may look like and the role played by database technology within it, and secondly, what the technical issues are in applying database technology which require further investigation ?

On examining the two application areas (Naval and TMD C^2) and the TDS demonstrator system currently being developed, it can be foreseen that the system resulting from the research programme is likely to have the following features:

- The core of the system performing the higher-level C^2 functions will be a large real-time knowledge based system.
- The KBS must be capable of assimilating a 'picture' of the events occurring in the external environment and responding within a time limit determined by these events. Generally, response times will be extremely short, therefore, the KBS must have a high performance reasoning capability, that is, solving a complex problem in a very short duration.
- The KBS will probably be based on distributed AI (DAI) techniques where several intelligent agents (each of which can be regarded as mini expert systems) work cooperatively in performing the problem-solving task.
- The static knowledge base of the KBS is likely to be extremely large and shared by the intelligent agents.
- The KBS will access data held in several geographically local or remote databases, ie. distributed databases.

- These databases will not only contain static (or fixed) data about geographic features, weapons logistics, etc.but also dynamic (or continuously updated) data originating from various sensor platforms such as radar.

From these system features, a number of issues arise in applying database technology which require further investigation:

- High performance: The C^2 applications real-time tasks where the response time is severely limited. Speed of accessing (retrieving and writing) the required data is a critical issue.

- Dynamic (or continuously updated) databases: Databases being updated with continuously changing data, ie.usually data originating from sensors.

- Management of large knowledge bases: The resulting KBS is likely to have an extremely large static knowledge base. Additionally, a distributed AI framework wants to have static knowledge sharable among and available to all its intelligent agents. A KBS with large static knowledge bases will probably hold much of this knowledge in secondary memory only fetching knowledge pertinent to the current problem being reasoned about into main memory. There is a requirement to be able to access the knowledge required to solve the current problem or sub-problem from a large static knowledge base as quickly as possible.

- Data integrity/consistency and DB maintenance: There is a need for human interrogation of the DB to check for DB integrity/consistency and maintain the data. This is mainly a system development issue.

2.6.3 Research Issues and Goals

Coupling

- Extension of RETE-like structures to large volume, volatile data environments.

- Tuning strategies for specific KBS shells being applied to given application requirements.

- Memory management strategies that integrate the primary / secondary access and the movement of data from the external database to internal KBS representation.

- Use existing vendor knowledge on KBS architectures and applications to 'kick-start' a clear approach to the limitations currently faced in KBS/DB coupling.

- Develop a theoretical model of areas in which performance is likely to be a problem with respect to different types of inferencing and storage structures.

AI-Enhanced Database

- Query Processor extension to include complex objects

- Query processor extension for non-First Order Predicate Logics

- Performance thresholds for Logic Programming based approaches

- Extension of relational model to complex objects : build upon DBMS vendors perception of future development

- Deductive Databases - fund academic research with a view to implementation details of proven concepts (include the often overlooked aspects of query processing, performance) and parallelism.

Intelligent Front-Ends / Natural Language

- Translating complex relationships in the data in a form they can be readily understood by the user.

- User-modelling as a technique for understanding the context of a query and therefore the most appropriate response.

- Management of distributed databases.

- Front-end query processor to decompose a query into a 'data' and inference component as appropriate.

- Draw upon practically-oriented academic research (eg. Glasgow) on approaches to information access.

- Get an over-view of current distributed database practitioners work and future plans.

- Look at models for translating representational forms into underlying logical structure

Novel Database Structures

- Large object problems

- Concurrency handling

- Strategies for getting objects out of secondary memory

- Get close to the application-oriented work that a number of OODB vendors are beginning to push.

- Tie into those academic groups that not only do work in the object-oriented field, but having a strong background in database work generally and are therefore better able to assess its applicability.

Dynamic Databases

- Time-constrained inference and search

- Assimilation of high volatility data patterns - 'bursts' of information

- Very limited research anywhere in implementing 'time-constrained' processing.
- It is necessary to 'bootstrap' this activity, using whatever pragmatic links can be struck. Reliance on 'real-time' expertise is unlikely to succeed, as this typically involves very ad hoc solutions.

Knowledge Base Management Systems

- Unified approach to data and knowledge, search and inference.
- Evaluate the very few attempts at large scale structural integration.
- Adopt a 'seed-bed' approach and fund a large number of promising theoretical components that are at some stage likely to generate applicative findings.

2.7 Development Methods

The Development Methods research stream is intended to provide the main demonstrator programme with a sound foundation for the development of KBS applications throughout the range of command and control applications in both the Naval Operations and TMD scenarios. It will address a number of key technologies, including KBS life-cycle, knowledge acquisition, verification and validation.

The research to be conducted is not perceived to be a key application technology, but rather will ensure that KBS applications can be both developed and deployed with confidence. Moreover, by providing adequate supporting technology throughout the KBS life-cycle it is our objective to ensure that the development of KBS within the main demonstrator programme can be conducted as efficiently and economically as is possible given the current generation of technology available.

2.7.1 Structure and Scope

In this section we identify the subtopics within the stream and give an indication of the overall scope of each subtopic.

KBS Specification

This sub-topic addresses all aspects of specification development as they apply to Knowledge Based Systems. KBS differ from other software in having a different development cycle, which generally involves successive prototyping, and usually tend to have only an ill-defined user requirement. This sub-topic is concerned with all aspects of the development of a KBS specification and covers both ad-hoc and formal approaches.

KBS Validation and Verification

This sub-topic is concerned with techniques and methods for the validation and verification of KBS. This sub-topics is closely linked with the KBS Specification sub-topic since some form of specification willl be required in order to validate the KBS.

KBS Life Cycle Model

A number of methods for controlling the development of KBS, or KBS life-cycle methodologies, are emerging from current KBS research into usage on real applications. This sub-topic is concerned with the application of KBS methodologies such as KADS to the development of KBS for command and control problems.

Knowledge Acquisition

This sub-topic addresses methods for eliciting specific types of knowledge found in command and control domains, such as spatial and temporal relations. The applicability of existing methods and tools to such knowledge acquisition problems is also a concern to this sub-topic.

KBS Development Tools

Tools are available to assist in the development, testing and maintenance of KBS. For example, systems that perform some rule-base consistency checks exist, as do ones that integrate knowledge acquisition and development processes. The applicability of existing tools and the viability of building domain specific ones in the command and control domain is the objective for this sub-topic.

Robust Architectures for KBS

Within conventional software engineering there are a number of techniques for ensuring the robustness of software through control of the design and validation processes. The applicability of these to KBS design is addressed under this sub-topic heading.

KBS Maintenance

Because of the types of function KBS are expected to perform in the command and control context, it is highly probable that there will be changes required in the knowledge bases they contain. These changes may occur over both short and long periods, for example changes to tactics are short-term, whereas changes to ship libraries will happen over longer timescales. This sub-topic addresses how knowledge bases can be designed for updating and how the process should be managed.

Machine Learning

Machine Learning has a potential impact in two areas: as an adjunct to other knowledge acquisition methods at the development stage, and as a mechanism for improving or expanding system performance, i.e. as part of a maintenance and updating process. The objective of this sub-topic is to investigate the applicability of current machine learning technology to these aspects.

2.7.2 Relevance to Goals and Objectives

The ultimate objective of ARE is to use the results of the CCRP to deploy operational advanced C^2 systems. KBS techniques will be a substantial component of these systems. To be certain this objective is met, the following requirement must be satisfied:

- To manage and monitor system development such that the final operational system fits the purpose and is reached in the most economical fashion (from both the aspects of finance and effort).

Huge amounts of effort have been invested in developing formal and rigorous verification and validation methods for C^2 systems employing conventional software techniques. However, little investment has been made for KBS techniques as most of the effort has concentrated on demonstrating feasibility rather than getting the system operational.

One of ARE's main reasons for adopting KBS techniques in the CCRP is to embed 'intelligence' within a C^2 system. In order to achieve an 'intelligent' capability, existing knowledge pertaining to the C^2 task needs to be embedded within the system during knowledge acquisition. If this cannot be achieved, then the use of KBSs in C^2 is questionable.

2.7.3 Research Issues and Goals

A number of key technical issues need to be addressed in order that the objectives of the Operational Deployment and TMD scenarios can be met. They are primarily motivated by the overall immaturity of KBS as a technology and the need to establish methods and techniques to enable complex systems to be developed in such a fashion that they can be: deployed with confidence, operated with confidence and maintained with confidence. The main issues identified are:

The central theme within the stream is the development of components of a KBS methodology specifically oriented towards command and control applications. The major exception to this is in the area of Formal KBS Specification where a limited amount of existing work has been identified and the problems are considered to be complex.

Given the overall funds available within the programme it would be unrealistic to expect that the Development Methods stream will be able to deliver novel technical solutions on a broad front. Instead it is our expectation that this research stream will utilise the results of other major research initiatives, which will already have established the basic technological foundations, and will focus upon the adaption and customizing of the results for more effective use in the command and control domain.

3 ANALYSIS AND EVALUATION

The research sub-topics within each Technology Stream have been evaluated and prioritized against the criteria described below, to assess their suitability for inclusion in the research programme.

It should be emphasised that the final decision about any research sub-topic must take into account dependencies with other sub-topics, in the same or some other Stream, which are not reflected in the evaluation criteria. For this reason, the evaluation results are supplemented by a statement of the supporting arguments and justifications for their role in the programme.

The evaluation procedure has been performed with respect to each of the three principle objectives :

<u>Operational Deployment (OpDep in tables)</u>

To achieve the deployment of Naval KBS-based Command & Control systems using the TDS as a baseline.

<u>Enhanced Functionality (EnFun in tables)</u>

To extend the functionality of current Naval KBS-based Command & Control systems using the TDS as a baseline.

<u>TMD Application (TMD in tables)</u>

To deploy and deliver the required functionality for TMD Command & Control systems using UKAS/BMC^3 as a baseline. Our judgement for TMD ratings has been based on very specific scenarios and a view of TMD as a comparatively constrained domain in relation to Naval or Air Defence operations.

3.1 Evaluation Criteria

A set of five criteria was used to perform the initial evaluation of the research sub-topics. The assessment of sub-topics against each of the criteria was carried out as follows :

3.1.1 Relevance

The relevance of the sub-topic in order to achieve the objectives of OpDep, EnFun and TMD.

Categories : High, Medium, Low

3.1.2 Criticality

The extent to which research in the sub-topic would be critical to the achievement of each of the objectives of OpDep, EnFun and TMD.

Categories :

Critical - the objective could not be met without further research in this sub-topic.

Significant - the sub-topic is of significant importance to the objective, and may turn out to be critical.

Desirable - the sub-topic is relevant to the objective, but not essential. Research in this area might result in a better solution to the technical problems, but could not be considered as critical at any stage.

3.1.3 Timing

In order to clarify the timings criteria, we need to introduce the MOD(PE) development life cycle. This life cycle comprises of the following phases:

1. Pre-feasibility Study
2. Feasibility study
3. Project Definition 1
4. Project Definition 2
5. Full Scale Development (resulting in operational deployment)

Enabling research feeds into the Pre-feasibility Study and Feasibility Study phases. The timing criteria are considered with this life cycle in mind, and the timing quoted is that required to demonstrate the feasibility of the technology. The following timing units are used :

Naval (OpDep and EnFun)

The timing units for the Naval objectives, OpDep and EnFun are:

Short	<3 years
Medium	3-6 years
Long	>6 years

where for OpDep the timing is to the start of the Project Definition 1 phase, and for EnFun the timing is to the start of the Feasibility Study phase.

TMD

To be reasonably compatible with Working Paper 1 timescales (1991 - 2003 as range for demonstration and 1995 - 2005 for definition of requirement), the following units for timings to demonstration of feasibility (start of Project Definition 1 phase) have been adopted:

Short	<5 years
Medium	5-10 years
Long	10-15 years

3.1.4 Technical Feasibility

The technical risk in exploring the technology. If a technology is highly feasible, then it is low risk and will yield results within realistic timescales.

Categories : High, Medium, Low

3.1.5 External Mainstream Research

A brief assessment of the amount of mainstream research being carried out external to the ARE, eg. IED, ESPRIT, etc.

Categories: High, Medium, Low.

3.2 Sorting Procedure

Following the evaluation of the individual sub-topics, they were then prioritized within Streams. The sorting procedure was carried out as follows :

Sort initially on the following criteria:

- firstly, criticality;
- secondly, technical feasibility;
- and thirdly, timing (good for timing implies short),

for each of the objectives - OpDep, EnFun and TMD.

The sorting results are tabulated and for each of OpDep, EnFun and TMD, numbers have been allocated against each sub-topic according to their priority ("1" being of the highest priority).

The sorting procedure is concerned mainly with the criticality and technical feasibility of the sub-topics, and does not take into account factors such as the levels of on-going research within the sub-topic.

3.3 Evaluation Results

3.3.1 Hardware Architectures

Summary Evaluation

Research Topic	Objective	Relevance	Criticality	Timing	Technical Feasibility	Relevant Ext Res
Paradigms	TMD EnFun OpDep	High High High	Sig Sig Sig	Long / Medium	Medium	Low
Tools	TMD EnFun OpDep	High Medium High	Crit Des Sig	Long / Medium	Low / Medium	Low
Architect-ures	TMD EnFun OpDep	High Low High	Sig Des Crit	Long Medium Short	Medium / High	High
Neural Nets	TMD EnFun OpDep	Medium Medium Medium	Sig Des Des	Long	Medium / Low	High / Medium

Sort Results

Sub-topic	OpDep	EnFun	TMD
Paradigms	2	1	2
Tools	3	2	1
Architectures	1	4	4
Neural Networks	4	3	3

Interpretation

Because of the strong inter-dependencies which exist between the first three sub-topics in this stream, it is extremely difficult to establish a set of rankings on them which are completely consistent.

Although paradigms and tools are both significant for OpDep, hardware is seen as critical, primarily because of its potential to offer an acceptable solution in the short term. (Note that it is anticipated that this task may not require extensive an "research" element). If this is not viable due to a design freeze, then paradigms should be investigated to see if they can offer the required increase in performance. If orders of magnitude increases in performance are required, then the rankings become much more like those for TMD.

The most important sub-topic for EnFun is paradigms. They are seen as being significant and highly relevant, whereas the other can only be viewed as desirable and of medium relevance. The high degree of external activity in Hardware research places it at the bottom.

The critical role played by tools in the TMD requirement, coupled with the low degree of external activity imply that this the most important topic for research, although the relationship with paradigms is very close, and it is unlikely that one would be studied without the other. Although hardware is more relevant to TMD than neural nets, as with EnFun, the high degree of external research in hardware means that neural nets may be a more pertinent sub-topic for this particular research programme.

The rankings for neural networks may be felt to be too low. If they were known to be a consistently successful technology whose past performance could be seen to be important to systems which had failed using other processing techniques, then they may be seen as more critical. However, the state-of-the-art in current research has shown that they do have relevance, and they should be viewed in some sense as a subset of paradigms, and therefore important in terms of achieving the aims of EnFun and TMD.

3.3.2 Systems Design

Summary Evaluation

Research Topic	Objective	Relevance	Criticality	Timing	Technical Feasibility	Relevant Ext Res
DAI	TMD EnFun OpDep	High High Low	Crit Crit Des	Medium Medium N/A	Medium Medium N/A	High
R-T Sys Eng	TMD EnFun OpDep	Medium Medium High	Sig Sig Crit	Medium Medium Short	High High High	Medium

Sort Results

Sub-topic	OpDep	EnFun	TMD
DAI	2	1	1
RT- Sys Eng	1	2	2

Interpretation

The TDS and TMD demonstrator programmes will be putting a lot of effort into various issues associated with real-time processing. This will include, for the TMD, bench marking to ascertain what performance improvements will be needed for real-time deployment. Until these results are available some of the difficult issues in real-time systems design, specifically the need for different reasoning strategies, may not be relevant.

3.3.3 Knowledge Representation and Manipulation

Summary Evaluation

Research Topic	Objective	Relevance	Criticality	Timing	Technical Feasibility	Relevant Ext Res
Temporal	TMD EnFun OpDep	High High Low	Crit Crit Des	Short Short N/A	High High N/A	High
Agent Modelling	TMD EnFun OpDep	Medium Medium Low	Sig Sig Des	Medium Long N/A	Medium Medium N/A	Low
Spatial	TMD EnFun OpDep	High High Low	Crit Crit Des	Long Long N/A	High Medium N/A	Low
Planning	TMD EnFun OpDep	High High Low	Crit Crit Des	Medium Long N/A	High Medium N/A	Medium
RMS	TMD EnFun OpDep	Medium Medium High	Sig Sig Sig	Medium Medium Medium	High High High	High
Uncertainty	TMD EnFun OpDep	High High Medium	Crit Crit Des	Medium Short Short	Medium Medium Medium	High
Situated Action	TMD EnFun OpDep	Low Medium Low	Des Des Des	Medium Medium Medium	Low Low Low	Low

Sort Results

Sub-topic	OpDep	EnFun	TMD
Temporal	3	1	1
Agent Modelling	3	6	6
Spatial	3	3	3
Planning	3	3	2
RMS	1	5	5
Uncertainty	2	2	4
Situated Action	7	7	7

Interpretation

The results look balanced, but there are factors which they do not take in to account. In particular, it must be borne in mind that a low rating in this table does not mean a topic should or can be ignored. The case in point is Agent Modelling which will be essential to the functioning of any planning system i.e. the inter dependencies will require that research is put in to some of the lower ranked topics.

3.3.4 Human Computer Interaction

Summary Evaluation

Research Topic	Objective	Relevance	Criticality	Timing	Technical Feasibility	Relevant Ext Res
Physical Interfaces	TMD EnFun OpDep	High High High	Sig Sig Des	Medium Medium Short	Medium Medium High	Medium
Design Methods and Tools	TMD EnFun OpDep	High Medium High	Crit Des Crit	Long Medium Medium	Medium N/A Medium	Medium
Modelling Issues	TMD EnFun OpDep	High Low High	Crit Des Crit	Medium Medium Medium	Low Low Low	Low
Cognitive Issues	TMD EnFun OpDep	Medium Medium Medium	Des Des Sig	Medium Medium Medium	Low Low Low	Low
User Support	TMD EnFun OpDep	Medium Medium High	Sig Sig Crit	Medium Medium Medium	Medium Medium Medium	Medium
Organisati-onal Issues	TMD EnFun OpDep	Medium Medium High	Des Des Sig	Medium Long Long	Low Low Low	Low

Sort Results

Sub-topic	OpDep	EnFun	TMD
Physical Interface	3	1	3
Design Methods and Tools	1	5	1
Modelling Issues	4	6	4
Cognitive Issues	5	3	6
User Support	2	2	2
Organisational Issues	6	4	5

Interpretation

The results of the sorting procedure would seem to indicate two groups of sub-topics. The sub-topics of Physical Interface, Design Methods and Tools and User Support form one group of roughly equally rankings; the sub-topics of Modelling Issues, Cognitive Issues and Organizational Issues form another group of similar rankings to each other but of lower priority overall to the first group of sub-topics.

The conclusion is therefore that the first choice of experiments should address the sub-topics of the Physical Interface, Design Methods and Tools, and User Support.

3.3.5 Database Knowledge Base Interaction

Summary Evaluation

Sub-topic	Objective	Relevance	Criticality	Timing	Technical Feasibility	Relevant Ext Res
Coupling	TMD EnFun OpDep	Medium High High	Des Crit Crit	Short Short Short	High High high	Low - Med
AI-Enhanced Databases	TMD EnFun OpDep	Medium Medium Medium	Sig Crit Des	Medium Medium Medium	High Medium Medium	Med-High
Intelligent Front-ends	TMD EnFun OpDep	Low High High	Des Crit Sig	Medium Long Short	Medium Medium Medium	High
Novel Database Structures	TMD EnFun OpDep	High High Medium	Sig Sig Sig	Medium Medium Short	Medium Medium Medium	Medium
Dynamic Databases	TMD EnFun OpDep	High High High	Crit Crit Crit	Medium Medium Medium	Medium Medium Medium	Medium
KBMS	TMD EnFun OpDep	High Medium Low	Crit Sig Des	Long Long Long	Medium Medium Medium	Medium

Sort Results

Sub-topic	OpDep	EnFun	TMD
Coupling	1	1	5
AI-Enhanced Databases	5	3	3
Intelligent Front-ends	3	2	6
Novel Database structures	4	6	4
Dynamic Databases	2	4	1
KBMS	6	5	2

Interpretation

The progression from current state of the art in Knowledge Base / Database interaction (coupling) through AI-enhanced database to structurally integrated KBMS is well represented in the matrix if the scenarios are taken to represent the time dimension (ie. KBMS is low in the Naval scenarios and high in the TMD).

Novel database structures have to be viewed as alternative ways of implementing data structures and therefore appear relatively low down the list due to the difficulty of classifying them as critical. This may overlook their ability to provide significant performance improvements rather cheaply.

The Dynamic Database work is justifiably classified above Intelligent Front End on all except the EnFun scenario, where it is relegated due to possible poor technical feasibility.

AI-enhanced appears low in all scenarios, perhaps artificially so. There is a good case for it making a significant contribution to the EnFun scenario, despite the fact that it is relegated by its 'Medium' feasibility.

3.3.6 Development Methods

Summary Evaluation

Sub-topic	Objective	Relevance	Criticality	Timing	Technical Feasibility	Relevant Ext Res
Formal KBS Specific.	TMD EnFun OpDep	Medium N/A Low	Sig N/A Des	Medium N/A Long	Low N/A Low	Low
KBS V&V	TMD EnFun OpDep	High N/A High	Crit N/A Crit	Medium N/A Medium	Medium N/A Medium	Low
KBS Life cycle Model	TMD EnFun OpDep	Medium N/A Medium	Sig N/A Sig	Short N/A Short	High N/A High	High
Knowledge Acquisition	TMD EnFun OpDep	Medium Medium Medium	Sig Sig Sig	Short Short Short	High High high	High
KBS develop-ment tools	TMD EnFun OpDep	Medium N/A Medium	Des N/A Des	Medium N/A Short	High N/A High	Medium
Robust KBS Architec.	TMD EnFun OpDep	High High high	Crit Crit Crit	Medium Medium Medium	Low Low Low	Low
KBS Mainten-ance	TMD EnFun OpDep	High N/A High	Crit N/A Crit	Medium N/A Medium	Medium N/A Medium	Low
Machine Learning	TMD EnFun OpDep	Medium N/A Medium	Des N/A Des	Medium N/A Long	Low Low Low	High

Sort Results

Sub-topic	OpDep	EnFun	TMD
Formal KBS Specification	7	N/A	6
KBS V&V	1	N/A	1
KBS Life Cycle Model	4	N/A	4
Knowledge Acquisition	4	2	4
KBS Development Tools	6	N/A	6
Robust Architectures	3	1	3

KBS Maintenance	1	N/A	1
Machine Learning	7	N/A	7

Interpretation

The N/A (Not Applicable) ranking is used to indicate that, for the most part, the issues raised by this stream are independent of the efforts to enhance functionality, and therefore do not play a part in ranking subtopics for that objective.

The results of the technology evaluation have identified three key areas which will need to be addressed in order to enable the deployment of KBS applications in both the OpDep and TMD application areas. Each of these substreams is considered to be both highly relevant and critical to both of these application domains since without an improved capability in each area it is unlikely that any KBS application will be considered to be sufficiently mature to be deployed.

The command and control applications in both the OpDep and TMD domains place requirements upon the confidence with which KBS can be deployed, operated and kept operational. These factors have influenced the relative importance of the substreams. The key areas identified areas are: Validation and Verification; KBS Maintenance and Robust Architectures for KBS.

4. Summary of Evaluation Results

The results of the evaluation of potential research sub-topics are summarised below. Within each Stream, the sub-topics are sorted and grouped according to their criticality for each objective (OpDep, EnFun, TMD) :

Critical

Significant

Desirable

Since the System Design stream only contains two sub-topics, consideration of these has been amalgamated with the Knowledge Representation sub-topics.

4.1 Hardware Architectures

Operational Deployment

Hardware
Paradigms Tools
Neural Nets

Enhanced Functionality

Paradigms Tools Neural Nets Hardware

TMD Application

Tools
Paradigms Neural Nets Hardware

The results of the evaluation of sub-topics within the Hardware Architectures Stream are based on the following assumptions :

- the performance required for the Naval system may or may not require a parallel hardware solution, but TMD systems almost certainly will.
- if a parallel solution is required, the emphasis in the research programme should be on the definition of suitable paradigms.
- the requirement for tools reflects the need to implement paradigms.

- some technology tracking of hardware options will be required, regardless of the type of option selected (parallel v. conventional).

- Neural networks are clearly relevant, but cannot be seen as critical at this stage.

4.2 Systems Design / Knowledge Representation

Operational Deployment

RT systems engineering
Reason Maintenance System
Uncertainty Distributed AI Temporal Agent Modelling Spatial Planning Situated Action

Enhanced Functionality

Temporal Uncertainty Distributed AI Spatial Planning
RT Systems Engineering Reason Maintenance System Agent Modelling
Situated Action

TMD Application

Temporal Planning Spatial Distributed AI
Uncertainty RT Systems Engineering Reason Maintenance System Agent Modelling
Situated Action

The results of the evaluation of sub-topics within the Real-Time Systems and Knowledge Representation & Manipulation Streams are based on the following assumptions :

- the current TDS is based on a single-chip architecture, and it would be inappropriate to re-design it as a distributed system for operational deployment. However, it is probable that later enhancements will be based on multi-processor platforms.

- Real-time systems engineering will be critical to the operational deployment of the naval system, but relatively little further research may be needed for EnFun and TMD.

- The knowledge-base architecture of the TDS has been defined and it would be inappropriate to re-design it using alternative knowledge representation schemes. However, the requirements of the enhanced systems are likely to exceed the capabilities of the current rule-based design and will necessitate research into more powerful representations of temporal and spatial reasoning, uncertainty and agent modelling.

- Planning is concerned entirely with enhanced functionality and has no relevance to operational deployment of the current TDS.

- Reason Maintenance systems may be useful for OpDep but in the longer term are likely to be overtaken by developments in the field of reasoning about uncertainty.

4.3 Human Computer Interaction

Operational Deployment

Design Methods & Tools
User Support
Modelling Issues

Cognitive Issues
Organisational Issues

Physical Interface

Enhanced Functionality

Physical Interface
User Support

Cognitive Issues
Organisational Issues
Design Methods & Tools
Modelling Issues

TMD Application

Design Methods & Tools
Modelling Issues

User Support
Physical Interface

Organisational Issues
Cognitive Issues

The results of the evaluation of sub-topics within the Human Computer Interaction Stream are based on the following assumptions :

- the physical interface for the current TDS is already well developed, and further research would not be critical to its operational deployment.

- modelling issues are seen as the most critical to the programme in the longer term, and are the area in which HCI research can probably make the most important contributions.

4.4 Database / Knowledge Base Interaction

Operational Deployment

Coupling Dynamic Databases
Intelligent Front ends Novel Databases
AI-enhanced Databases KBMS

Enhanced Functionality

Coupling Intelligent Front ends AI-enhanced Databases Dynamic Databases
KBMS Novel Databases

TMD Application

Dynamic Databases KBMS
AI-enhanced databases Novel databases
Coupling Intelligent front ends

The results of the evaluation of sub-topics within the Database / Knowledge Base Interaction Stream are based on the following assumptions :

- the direct coupling of existing database and knowledge base components is the most pragmatic approach in the short term, and therefore the most appropriate for OpDep. In the longer term it will probably be overtaken by developments in novel databases.

- research into the design of dynamic databases will be critical to all three objectives.

- the timing of research into AI-enhanced databases and Intelligent front ends will be too long to benefit OpDep.

- at present, it is not clear how critical the developments in novel databases will prove to be. However, there may be significant performance benefits from this approach.

- in the long term, KBMS's may have an important impact on the problems of knowledge base maintenance.

4.5 Development Methods

Operational Deployment

Validation & Verification KBS Maintenance Robust Architectures
Life-cycle model Knowledge Acquisition
Development tools Machine Learning Formal Specification

Enhanced Functionality

Robust Architectures
Knowledge Acquisition

TMD Application

Validation & Verification KBS Maintenance Robust Architectures
Life-cycle model Knowledge acquisition Formal specification
Development tools Machine Learning

The results of the evaluation of sub-topics within the Development Methods Stream are based on the following assumptions :

- the problem of specifying knowledge based systems is of key importance to the procurement process and is therefore of paramount importance in the research programme.

- the Formal Specification sub-topic relates only to the application of mathematical specification techniques, such as 'Z' and VDM. Alternative approaches to specification are explored in the Validation & Verification sub-topics.
- knowledge acquisition is central to the development of KBS, but is still under-researched, particularly for the more complex knowledge representation formalisms.
- KBS development tools are not seen as critical to any of the objectives, and currently available tools may suffice.
- machine leaning is clearly relevant to the programme but cannot be regarded as critical at this stage.

Annex B1 : Technology and Application Streams

Table of Contents

1. HARDWARE ARCHITECTURES

1.1 State of the Art

Paradigms

There has been an increasing interest in the last ten years in the development of systems capable of using many different sources of sensor information (Faugeras 1986, Henderson 1987, Durrant-Whyte 1987b, Porrill et al 1987, Miles 1988). This interest arises from a realisation that there are fundamental limitations on any attempt at building descriptions of an environment based on a single source of information: a single sensor can only ever provide partial information about an environment, insufficient to arrive at a single interpretation and limited in resolving ambiguity. Diverse information from many different sources can be used to overcome the limitations inherent in the use of single sensors, by providing different additional constraining information, by providing redundant information to test and verify partial sensed hypotheses (or select between multiple hypotheses), and by improving the overall robustness of the sensing system (Giralt 1984, Henderson 1984).

The remainder of this section is organised as follows. Firstly, the technical background to the work is given, following which existing results and ideas in terms of multi-sensor fusion are established by reference to published literature.

The sensors of a multi-sensor system are diverse and logically distinct. Sensor measurements are often imprecise and frequently unreliable. Thus single sensor, single algorithm systems are limited in their ability to resolve ambiguities, identify spurious information, or detect errors. Such failings are an unavoidable consequence of attempting to make global decisions on the basis of incomplete, local, or underconstrained information. The motivating goal behind all sensor integration techniques is to actively utilise the diversity of information available from multiple sensors in overcoming the limitations of any one sensor system.

There is a commonly accepted hierarchy in the functionality of a sensor integration scheme:

Data Fusion: Data acquired from a number of similar or different sensors is fused to provide a single coherent model of the world.

Interpretation: This model is then interpreted in terms of "meaning", i.e. labels are attached to objects, events are recognised, etc.;

Situation Assessment: The meanings attached to the data are analysed for significance- is this a normal situation? has an error occurred? etc.;

Resource Allocation: Once a situation has been assessed, it is necessary to determine the response of the system to that situation, e.g. active sensors could be deployed in a region of great activity.

Note that as progress is made up this hierarchy there is a continual increase in the level of abstraction, and typically a move away from numerical representations and towards symbolic ones. Also, the final level involves planning, and as a consequence a downward flow of control.

It is natural that such a hierarchical paradigm should be mirrored by a hierarchical implementation: this is precisely the case with many fusion systems (Harmon 1986, Shafer and Thorpe 1986). However, although hierarchical, this paradigm is not necessarily centralised. Nevertheless, most theoretical studies, and nearly all practical systems, make critical assumptions about the degree of centralisation that is either necessary or desirable. From a pragmatic point of view, centralised systems are easier to build and then study than are distributed ones. Theoretically, it has until quite recently been unclear that decentralisation could be achieved without loss of optimality. Recent work in has shown, however, that full decentralisation of an Extended Kalman Filter (EKF) multi-target tracker is achievable (Durrant-Whyte et al 1989), which leads us to believe that similar results may be obtained in more complicated situations.

To be able to fuse information obtained from a number of different sensing devices, three things are required: a common language, a good model of sensor performance, and an efficient architecture.

However, we can identify several important sub-problems within data-fusion which provide a useful framework for this discussion. In particular, multiple sources of information raise the following generic questions:

- A consistency test must be applied to the various data. That is, do they relate to the same physical entity? This is most commonly achieved through the use of statistical hypothesis tests, where the null hypothesis is that the data are consistent. The basis for evaluating the null hypothesis could be "similarity" (e.g. confidence of matching), "distance" (e.g. Euclidean), "likelihood" (e.g. chi-squared test), or other measures derived from the error model.

- The data are then combined according to some policy or rule e.g. maximum likelihood, minimum error, Extended Kalman Filtering (Ayache and Faugeras 1988), fuzzy logic (Zadeh 1983), or Dempster-Shafer theory of evidence (Garvey and Lowrance 1981). More specifically, Kalman filtering is one of a number of techniques developed in the control literature, and has been successfully applied to the fusion of low-level geometric primitives which have well-defined associated error characteristics (Faugeras and Ayache 1986). Since the Kalman filter requires affine subspaces (points, lines, planes), it is difficult to apply to the fusion of symbolic information such as "green objects are trees".

- A strategy must be used to control the cooperating sensors (Shafer and Thrope 1986, Durrant-Whyte 1988a). This could be:

  (a) competitive: typically used for sensors producing the same type of data, such as two cameras with overlapping fields of view;

  (b) complementary: applying each sensor individually where appropriate, e.g. detecting an object with a low resolution sensor and then using a high resolution device for identification purposes;

Hybrid strategies are also possible. The sensors of a multiple sensor system can be considered as members of a team (Hager 1988): each team member observes the environment and makes local decisions, each contributing to a consensus view of the world, and cooperating to achieve common goals. A multi-sensor system can therefore be viewed as a team of Bayesian estimators, communicating through the common language of uncertain geometry (Durrant-Whyte 1988b).

Generic solutions are especially important because application domains are so varied, and place such a wide range of requirements on any "practical" solution.

There are essentially two different approaches to the data fusion problem, that based on the use of quantitative data fusion algorithms, or that based on the use of qualitative modelling and fusion methods. We shall briefly survey the main data fusion methods and highlight some of their advantages and disadvantages. We advocate that effort should be concentrated on integrating qualitative and quantitative fusion (IQ^2 fusion) techniques, and suggest two methods which are proposed for further investigation.

Quantitative Methods

Quantitative data fusion methods rely on the fact that the information obtained from sensors can be described in a purely numeric way. Given this quantification of sensor data, common parametric methods can be used to describe and communicate information, and a wide range of statistical and decision theoretic methods can be employed to fuse information. Most techniques involve the use of linear decision rules, such as the Extended Kalman Filter (Gelb 1974), to manipulate and combine information, although some non-linear methods are also now being used (Hager 1987, Hager 1988). The principal advantage of quantitative fusion techniques is in their use of well-understood methods of statistical modelling (Box and Jenkins, 1976), parameter estimation (Berger 1985), and linear filtering (Bar-Shalom and Fortmann, 1988). The most popular quantitative data-fusion technique is geometric fusion, of which image-based fusion is the least abstract example.

Geometric fusion methods are motivated by the view that sensors can be regarded as "geometry extractors": sources of partial, uncertain, geometric information about a sensed environment (Durrant-Whyte 1987b). The most important aspect of these techniques lies in their explicit use of geometry as a model of information (Ayache and Faugeras 1988, Bolle and Cooper 1986, Porrill et al 1987). This provides a common language for the communication of information between different sensors, and allows both the tools of formal geometry and of statistical decision making to be utilised in the processing, integration and interpretation of sensor information (Durrant-Whyte 1988a). The principle advantage of using geometry to model sensor information is that it allows all types of sensors to be considered in a common framework, enabling the development of efficient fusion algorithms. However, this explicit use of geometric representation also gives rise to the primary disadvantage of these techniques: the imposition of specific geometric descriptions of information results in representational and heuristic inadequacy, restricting the application of such methods to relatively well-structured, geometric, environments.

Image fusion techniques integrate information by direct fusion of intensity arrays, obtained from electro-magnetic scanning sensors such as CCD arrays for example (see Mitche and Aggarwal 1986 for a survey of these methods). They are generally applicable to the problem of integrating information from different imaging sensors, such as vision and passive infra-red (Nandhakumar and Aggarwal 1986). Their main advantages stem from the direct usage of tried and tested vision algorithms, and the fact that they place no constraint on representation or interpretation of sensed data (Marroquin 1985). Their primary application area is in image-enhancement and image-based feature detection. Their outstanding disadvantage is their inability to deal with sensor information which is not intrinsically image-based. This precludes, for example, the use of contact or manipulation information from tactile or force sensors. In addition, most image fusion research addresses the fusion of images with similar resolution and close observation points (stereo).

Qualitative Methods in Sensor Fusion

The use of knowledge bases and logical methods to characterise and fuse sensor information has grown in popularity over the last five years. Logical, or knowledge-based integration methods embody the view that sensors can be considered as sources of knowledge about the structure of an operating environment (Henderson and Hansen 1985, Harris and White 1987, Byrne et al 1989). The key element in these techniques is to abstract the physical sensing process in terms of the information or knowledge they provide (Flynn 1985). This then allows the processing, interpretation and, most importantly, the control of information to be independent of the methods used to extract this knowledge. The first significant work in this area was the use of frames to describe the type of information available from the sensors of a process plant (Fox et al 1983). This system is principally of interest for its qualitative description of sensor failure modes. In a series of papers, Henderson (Henderson et al 1984, Henderson and Hansen 1985, Henderson et al 1988) developed the idea of a "logical sensor system", in which both sensors and algorithms were described by logical descriptions of function, input and output. These descriptions could be manipulated to construct a system for a given sensing task, but lacked any deep model of sensor capabilities. More recent work in this area has included the use of knowledge-based systems for track identification in naval warfare applications (Harris and White 1987, Byrne et al 1989).

The principal advantage of these methods are a consequence of this abstraction; because the sensors are described in terms of information, a natural form of knowledge redundancy, and not physical redundancy, can be used (Henderson et al, 1984). This is important because in seeking to provide a robust description of a sensed environment, it is the information and its interpretation which is important, and not always the physical sensing process. The primary disadvantages of knowledge-based integration methods are implicit; there is no obvious common language that should be used to communicate information, the methods employed to fuse the "knowledge" obtained from the sensors are rather ad hoc, and the mechanisms, organisation and use of these systems are poorly understood.

Combining Signals and Symbols

Both qualitative and quantitative data fusion methods have limitations, and neither type of technique is ideal in all situations. Thus, continuing this information integration philosophy at a higher level, these sensor fusion techniques should themselves be integrated to overcome the limitations of each individual technique. Specifically, quantitative integration techniques would benefit from the use of knowledge-based methods to organise, interpret and utilise the information they acquire, providing intelligent pre- and post-processing of sensed data, tailored toward individual sensing operations. Qualitative integration methods could benefit from being able to utilise statistical and parametric methods to fuse and describe low-level information. Current sensor fusion methods have not yet reached this level of sophistication. This is primarily because there is no universal way of representing the accuracy and uncertainty of the data, and it is not yet clear how probabilistic and rule-based information can be integrated in a coherent and mutually-beneficial manner. Rule-based programming methods are an important tool in overcoming the complexity inherent in multi-sensor systems (Harris and White 1987), but to be useful it is essential that they are able to utilise and reason about the many powerful statistical sensor-integration techniques currently in use; distributed estimation (Hashemipour et al, 1988) or data-association and target-tracking methods (Bar-Shalom 1978, Chong et al, 1986), for example.

Distributed and Decentralised Sensing

The complexity of the data-fusion problem has given rise to a considerable interest in the development of suitable architectures for multi-sensor systems. Hierarchical organisations have proved popular (Orlando 1984) because of their ability to hide the complexity of low-level processing with higher level computational layers. Blackboard architectures have also been extensively employed in multi-sensor systems (Shafer and Thorpe 1986, Harmon et al, 1984, Durrant-Whyte 1987a), because of an apparent ability to modularise sensor competences in terms of a number of communicating "agents".

The motivation behind all these organisations is to hide complexity and make each sensor function as modular as possible. Typically, in these organisations, the architecture comes first, and the algorithm is then "designed to fit". This often leads to severe problems in communication between different modules, and in the overall control of the system : who talks to whom, when and about what. Such problems have resulted in these organisations being controlled by a central processing facility - either to take care of high-level fusion functions, or to serve as a communication medium. This goes against some of the original aims of these architectures; the sensing modules cease to become "autonomous agents" and the central processor becomes a communication and computational bottle-neck. Centralised architectures are based on the idea that sharing the compute resource will increase efficiency. However, such architectures result in complex control structures often dedicated to particular applications.

The aim of fully decentralised sensor architectures is to avoid many of these problems by embedding as much processing expertise locally with each sensor as possible, and completely distributing both computation and communication. Decentralised architectures typically consist of a number of distributed sensing and decision making nodes each with sufficient processing power to undertake local signal processing and local decision making functions. To place local "expertise" at each node each node must be capable of building models, not just of their own function, but also of the other sensors in the system, and their relation to the environment itself. The goal of complete decentralisation is never likely to be achieved, nor is it necessarily desirable in all situations. However, the clear advantages of decentralisation, in terms of being able to build modular, scalable and robust sensing architectures, motivates the question of how much of the sensing and data-fusion process can be decentralised in this way.

The Goals of Decentralisation

Most existing multi-sensor systems are physically distributed but functionally centralised (e.g. the TDS). The difference between distributed sensing and decentralised sensing is that in the latter type of system the processing and interpretation of sensor information as well as the physical sensing device is distributed amongst the sensors themselves and does not take place in any central processing facility, i.e. distributed perception. The potential benefits of decentralised sensing systems over hierarchical, centralised architectures is that individual sensors can be produced in a modular manner, reliability can be improved through local detection of error, centralised processing bottlenecks can be eliminated through distributing processing, and systems can be built which are both flexible and scalable. The goal of decentralised sensing systems is to realise these potential benefits by first developing algorithms which will permit the distribution of fusion and decision making, and second by constructing an architecture which distributes as much processing power as possible, locally, with each sensor.

Decentralisation Methods

The advantages of being able to distribute processing and expertise, locally, amongst the sensors of a multi-sensor system has resulted in a considerable amount of work being conducted in problems of decentralised decision making and data fusion. This work broadly divides into three types: decentralised tracking and event fusion, multi-Bayesian methods, and distributed problem-solving. These areas are discussed below.

Decentralised Event Fusion

The detection, identification and tracking of events by a variety of different sensors is fundamental to the type of applications addressed by this research programme . Event fusion using information from a number of distributed sensors has been a major issue in a number of data-fusion applications but in particular military systems.

The first attempts at distributing some of the data fusion problem amongst a number of physically distributed sensor platforms involved the development of so-called track-to-track (TT) fusion algorithms (Singer and Kanyuck, 1971, Bar-Shalom 1981, Miles 1988). In the TT fusion problem, a number of sensors generate event information based only on their local data. At periodic intervals, this track information is sent back to a central processing facility where it is fused with tracks from other sensors. The limitations of this approach are clear; first, a considerable amount of central processing still needs to be performed (particularly in the association phase), and second, the sensors themselves lack any information about what other sensors can see and so can become quite inefficient in detecting outliers or spurious events (thus limiting the appeal of the data-fusion paradigm).

Researchers have also begun to look at algorithms designed for specific distributed sensing architectures. In particular, event tracking mechanisms were developed for hierarchical architectures (Chong 1979, Hashemipour et al 1988), which employed an explicit layering of TT fusion and data association mechanisms. These algorithms permitted the distribution of standard multi-target EKF methods in a strict architecture, while guaranteeing globally optimal performance.

Recently however, these results have been extended to allow complete decentralisation of multi-event tracking algorithms, first in a sub-optimal decomposition (Chong 1986) but now in a globally optimal manner for linear fully-connected networks of sensor modules (Rao and Durrant-Whyte 1989, Brown et al 1989). The fully decentralised event-fusion algorithm is capable of distributing a multi-target EKF data-fusion algorithm amongst an arbitrary number of physically distributed sensing sites, and arriving at a globally optimal solution to the event/track fusion process.

Multi-Bayesian Methods

There has been considerable work in the last thirty years on the use of probabilistic approaches to decentralised decision making. Much of this work has been undertaken in the field of theoretical economics in which two or more decision makers must communicate beliefs to arrive at a common consensus. Early work considered the extension of (antagonistic) game-theoretic methods to cooperative problem solving. In particular, the bargaining problem developed by Nash (Nash 1950) showed that under reasonable assumptions of rationality, two decision makers could arrive at an agreed decision without any subjective knowledge of what value each other placed on their respective information.

An extension of the two-person bargaining problem to multi-person agreements precipitated the development of team theory (Marshak and Radnor, 1972). Although still using a utility framework to compare decisions, team theory differs from standard bargaining problems by imposing considerably more structure on decision makers and the decision making process. In particular, the idea of an information structure has been employed to model the information available to individual decision makers. Team theory has seen considerable application in distributed control problems (Ho and Chu, 1972, Ho 1980).

Group decision making using purely probabilistic arguments to compare decisions has also been investigated. Early work by DeGroot (Degroot, 1974) showed how two decision makers could iterate to a consensus using a common Bayesian dialogue and showed how disagreements between decision makers might arise. The static multi-Bayesian system has since been extended to show how differing expert opinions might be combined (Morris, 1977), and how multiple Bayesians might integrate in explaining local, partial observations into common models (Bacharach, 1975, Bacharach, 1979). Dynamic multi-Bayesian systems have also been employed to investigate problems of distributed hypothesis testing (Weerahandi and Zidek, 1981, Weerahandi and Zidek, 1983).

Many of these results have found application in decentralised sensing problems. The results on agreement and disagreement in the bargaining problem have been employed in algorithms which seek a consensus amongst a number of geographically distributed sensing systems (Hager and Durrant-Whyte 1988). The idea of information structure as a model of sensor capabilities has also been used (Durrant-Whyte 1988a). Most important though has been the recent development of multi-Bayesian paradigms as a rigorous basis for sensor systems to communicate and integrate widely varying views of a common world (Hager 1988).

Distributed Problem-Solving

Distributed problem-solving is a very active research area in AI (e.g. Durfee et al, 1987, Bond and Gasser, 1988, Hautin and Vailly 1986, Gasser et al, 1987). However, most studies up to now have been concerned with distribution over software modules (abstract reasoning systems), as opposed to taking into account the paradigm of multi-agent systems where agents are concrete physical entities (Cammarata and McArthur, 1983, Chandrsekaran, 1981). In the former case, modules, even if implemented on a distributed hardware, do not have to face significant limitations in communication bandwidth. Shared memory architectures (e.g. Blackboards (Lesser and Corkill, 1979, Lesser and Corkill, 1981, Nii 1986, Engelmore and Morgan, 1988) are feasible solutions that have been found appealing in interpretation tasks (modules do not have to know about each other).

This approach may not apply to the multi-agent case: limited communication between agents may be necessary, and hence modelling and reasoning on others is required (Smith and Davis, 1981, Davis and Smith, 1983). Moreover, the sensor-agent in our case may have to react to what it perceives in the environment in order to pursue its tasks: it may have to change its position or some other functioning modality, adapting itself locally with some degree of autonomy to local changes. The sensor itself and its behaviour are in turn perceived by other agents as part of the environment. The multi-agent paradigm, where agents are physical entities, permits a greater modularity at the computation level (fusion and interpretation), but also at the control and (local closed-loop) adaptation levels.

The knowledge an agent has of itself, of its environment, and of other agents is not necessarily factual and valid: one speaks about the "belief" of an agent. The representation of such belief is a key issue in the design of a multi-agent system since it determines to a large extent the intelligent behaviour of each agent and the performance of the system.

Another major problem is how to maintain (or retrieve) the consistency of an agent's beliefs. One has to take into account the "internal revision" of belief, inherent to non-monotonic and hypothetical reasoning (which can be dealt with through TMS or ATMS-like approaches (deKleer 1986, Martins and Shapiro 1988)). More important in a distributed sensory system is the revision due to additional information perceived by the agent itself or sent by other agents. This fusion process, or combination of additional informations with current beliefs, is more inductive than purely deductive. It should rely on uncertainty models, that can be quite different for perceived and received data, and may depend on the reliability and dependency of the different sources.

Global interpretation has also to rely on and to manage the dialogue between agents that will take place to cope with conflicts, and more generally to deal with mutual knowledge (Durfee et al 1987). This dialogue is essential for addressing the problem of interpretation inaccuracy, that is the discrepancy between an interpretation (although locally consistent) and the real world.

Communication problems in a multi-agent system address not only the issue of common languages and protocols but also that of modelling and reasoning on communication acts as particular actions that change the state of the system (belief, knowledge, local and global interpretations) and enable cooperation (Rosenchein 1982). An agent's perception plan should make use of communication acts (e.g. move in that direction, request relative positioning from i to j, acquire data and send to k, that part of the interpretation meaningful for it), this is exemplified in the KAMP communication planning system (Appelt 1982) that rely on a NOAH type plan generator. A main issue here is the reference problem: besides geometrical and temporal features for localising objects in space and time (with explicit inaccuracy models), other descriptive characteristics are not necessarily meaningful for, and shared by, all agents. Attributes of the models and representations used by others may also need to be known to permit intelligent communication.

<u>Parallel Approaches to Knowledge Based Systems</u>

It is surprising that given the recent opportunity to exploit potential offered by parallel processing, surprisingly few examples exist of research into parallel approaches for rule based systems. There a number of reasons exist for this.

- First, attention has tended to focus on the use of parallel processing on problems which lend themselves well to parallelism (e.g. low level signal processing, CFD). Initially the focus was centred on where the need is to apply a number of identical operations to many pieces of dissimilar data (SIMD), and gave rise to machines such as the DAP. Recent developments have enabled these ideas to be extended to applying many different operations at the same time to different data items (MIMD) as embodied in the use of transputer arrays.

- KBS technology is still relatively immature; most "successful" versions of KBS implementation have by the very nature of the successs, been able to perform in suitable time frames.

- Parallel solutions to KBS may call for a radical departure from current thinking, and so have remained a topic of primarily academic research.

Having said this, there has been work on a number of topics which relate to this field, and which offer an indication of the potential for achieving some success in future work. There is the work carried out at ARE, which has aimed to extract the static and dynamic parallelism which may exist within a particular rule base, this has already shown that exploiting dynamic parallelism is likely to offer more potential for increased perfrormance than static parallelism (Daniel 1989). At a slightly more generic level, recent work in the U.S. has shown the possibility of implementing aspects of the RETE match algorithm in parallel (Miranker 1990), this is just one way in which parallelism can be exploited for production systems (Gupta 1987, Schreiner 1987).

There has also been a range of work which has looked at the use of parallelism in running applications built using conventional AI languages (Weening 1989, Trehan 1989, Gregory 1987). Unfortunately, this work is often characterised by illustrations that some form of parallel execution can be achieved for these languages, but it does little to suggest ways in which problems which are to be solved using these languages can actually exploit that parallelism. However, research along these lines has led to the recent introduction of tools which offer the potential for implementing parallel solutions to AI type problems more easily (Foster 1989).

It is this apparent lag of the work in solution techniques and tools to exploit and test parallelism, rather than the availability of parallel hardware on which to implement a parallel solutions, which will focus the needs of this research stream for this area. Decisions regarding hardware will need to be taken at some stage, and the technology is developing at such a rate that it is difficult to predict what will be available within the longer timescales of the programme. The currently available range of systems is described below.

<u>Parallel Machines</u>

State of the art parallel machines can broadly be split into multiprocessors which support parallel programming through shared memory space; multicomputers, supporting multiple SISD processors via message passing among distributed processors each with local memory and massively parallel machines which consist of a large number of processors operating in a SIMD or MIMD fashion.

Multiprocessors

Examples of this form of hardware include:

- BBN Butterfly, a machine with up to 256 processors (68020 and 68881 fpu). Each processor has local memory (4Mb) and has access to the total system memory via a butterfly switching network. Access times for network memory is typically 3 times that for local memory.

- Alice. This machine, based on transputers, comprises processing nodes, memory modules and a switchable interconnection network to allow the processors access to memory. The architecture is designed to allow efficient implementation of graph reduction tasks.

- Supercomputers. Many modern day supercomputers, such as the CRAY, use multiprocessing to achieve higher throughput.

Multicomputers

This is a rapidly expanding class of machines, examples of which are:

- Transputer arrays: Many makes of transputer arrays exist, the best known (Meiko, Parsys etc...) all have software controlled interconnection schemes. Transputers are powerful processors with 4 communication ports on chip, which allows arbitrary sized arrays to be built.

- iWARP: This machine is based on Intel/Carnegie Mellon devices. The machine is configured as a linear array of up to 72 cells. Each cell has a potential speed of 16MFLOPS, giving a total of 1.152GFLOPS.

- iPSC: This is a series of Intel based computers with a current top range system (iPSC/860) containing 128 processing nodes. The architecture is a hypercube of processors connected together with channels capable of 1.4MBytes/sec.

Massively parallel

This is a relatively small area of machines, mainly due to cost.

- Connection Machine: Originally conceived at MIT, a 65536, 1-bit processor array connected together in a 12 dimensional hypercube. This is a SIMD processor and was originally designed for concurrent manipulation of knowledge.

- AMT DAP: Designed and manufactured originally by ICL in the early 70's, this is an array of 1-bit SIMD processing elements, with local memory, connected together in a regular 2-D array.

- MasPar: This is a custom SIMD array with a maximum configuration of 16384 processors. Memory is distributed locally, but bi-directional data sharing is possible with a global communications router.

- NCube 2: This machine, with a maximum of 8192 processors uses a fine-grained MIMD architecture to achieve a claimed 19GFLOPS.

<u>Neural Networks</u>

Neural Networks have recently enjoyed a dramatic increase in worldwide interest. The state of the art in the field is changing dramatically on a yearly basis. What is yet to be determined is whether these techniques will be able to contribute positively to real world problems of significant size. Major successes to date in the field have tended to be in the areas of speech and vision. Problems which need a large amount of unstructured data to be processed or interpreted into a higher level of representation seem to offer the best application domains for neural networks. It is hard to judge whether or not there have been any "generic" results which can be quoted for a particular neural network across a range of the domains. The critical part of applying a neural network to a potential problem is finding an appropriate representation of the domain in such a way that the neural network can be applied. Because of this limitation, work which has been done in neural networks which is specifically aimed at the problems related to data fusion and target identification is of most relevance.

ARE are already leaders in this field, with their sponsorship of work carried out at Aberdeen University (Whittington and Spracklen 1989). This work in turn cites all of the major references in the field, e.g. Kuczewski 1987, Addison et al 1988).

Genetic Algorithms

Genetic Algorithms represent a specialisation of the more general area of stochastic optimisation. There are many potential applications for results from the area of stochastic optimisation within the general C^2 problem domain. For example: optimally combining observations, finding minimum risk allocations of defence resources, computing optimal deployment of assets, and performing threat assessment on incomplete data.

Work to date in the area of stochastic optimisation has concentrated on two main techniques: simulated annealing and genetic algorithms. However, the fact that only two algorithms are used belies the underlying complexity of the area of research. A good introduction to the general method may be found in the May 1989 issue of Artificial Intelligence. Stochastic optimisation methods are specifically designed to overcome optimisation problems which have been designated as NP-hard (Garey and Johnson 1979). Such problems, although easily soluble on a small scale, increase in computational complexity in such a way that they become computationally intractable once they reach a realistic size. The amount of CPU time required to "solve" the problem increases exponentially with respect to increases in the problem size. Stochastic optimisation methods tackle these problems by, for example a "guided" random walk through the solution space.

NP-hard problems typically occur in non-linear multi-variate optimisation, e.g. minimising the errors in multiple target tracking and identification, where the aim is to find a solution which minimises some given cost function. Furthermore, the cost function is defined over a very large discrete configuration space, and the concepts embodied in gradient descent methods may have no meaning. The guided random walk technique loosely consists of:

- attempting to find the best local minimum - this may be achieved by randomly changing the configuration (ie mutation), and by allowing new configurations to "inherit" desirable properties of their "parents".

- keeping improved results - allowing only the fittest children to survive, or the fittest of both parents and children.

- avoiding being trapped in local minima - this may be done by occasionally keeping worse results, ie by allowing less-fit mutations to occasionally survive.

This illustrates three characteristics of genetic algorithms; they often operate by using processes of reproduction, cross-over, and mutation. Indeed, the goal of genetic algorithm research is to improve the robustness (the balance between efficiency and efficacy) of the optimisation techniques available for NP-hard problems. Thus it is possible to characterise genetic algorithms as follows:

- they work with a coding of the parameter set, and not the parameters themselves;

- they operate on a population of solution vectors rather than a single one;

- they exploit values of the objective function directly, rather than derivatives of it or other auxiliary knowledge;

- they use probabilistic rather than deterministic transition rules.

A thorough review of genetic algorithms may be found in Goldberg (1989), which also contains a comprehensive bibliography.

1.2 Activities and Resources

1.2.1 Paradigms

UK Activity

- ARE
- Southampton University - C Harris
- Other MOD establishments, Defence Companies

Most UK activity in the data fusion area is focused in the military arena, with the notable exception of the work of Hugh Durrant-Whyte on robot vision at Oxford University.

Other Activity

- ESPRIT I - SKIDS

Availability of Resource

UK activity outlined above.

1.2.2 Tools

UK Activity

- A I Limited (STRAND Language), Imperial (PARLOG)
- Inmos, Meiko (Debugging Environments for the Transputer)
- Cambridge Consultants Ltd. (MUSE), Systems Designers (Real Time Systems)

Other Activity

- SUPERNODE II (ESPRIT II - Operating Systems and Programming Environments for Parallel Computers)

Availability of Resource

UK activity outlined above.

1.2.3 Hardware

UK Activity

- FLAGSHIP (Alvey Project on Parallel Processing)
- Imperial College (Novel Architectures for Parallel Processing and Neural Networks)
- Inmos, Meiko, BAe and Numerous Others (Transputer Developments)
- Brunel Uni., Bristol Uni., UCL (Content Addressable Memories)

Other Activity

- REX (ESPRIT II - Reconfigurable and Extensible Parallel and Distributed Systems).

1.2.4 Neural Networks

UK Activity

- The RIPR initiative (at RSRE Malvern) (D.Bounds, J. Mayhew)
- Oxford (I. Paige), Aberdeen (Prof. Spracklen), Edinburgh, Imperial College (Prof. Alexander), Cambridge, and numerous other Universities
- British Telecom (the CONNEX Programme)

Other Activity

- ANNIE (ESPRIT II - Establishing Industrial Standards for Neural Networks) Distributed Systems) - B.Ae. SRC, AEA Harwell
- Pygmalion (ESPRIT II - N.Nets for speech, vision, and tools for N.Nets) - Prof. Trelevan, UCL

Availability of Resource

UK activity outlined above.

1.2.5 Genetic Algorithms

UK Activity

- Strathclyde University (image processing, panning and scheduling)
- Bath University (fundemental research on novel methods of optimisation) - Prof. Silverman.
- Cambridge University (recovery of information from corrupted data) - Skilling and Gull, (non-linear optimisation) - Prof. Powell.

- Edinburgh Univeristy (optimisation of parallel algorithms on to Transputers),
- Oxford University (parallel algorithms, computational complexity, non-linear optimisation) - McColl, Sanders, Parrot.

<u>Availability of Resource</u>

Bath University

2 SYSTEMS DESIGN

2.1 State of the Art

2.1.1 Real Time AI Systems Design

Work in real-time AI issues tends to be application specific and "engineering" rather than "theory" oriented. Consequently there are not many clear examples of relevant work having been done. A survey article on Real-Time Artificial Intelligence (Laffey et al 1988) gives a good overview of existing recent work and shows the paucity of significant theoretical research into the issues raised above. The example projects cited by the authors do not include any in Command and Control, the closest being signal interpretation.

Overlooked in the above survey, but nevertheless some of the best and most relevant, is work being done by Victor Lesser and colleagues on the Distributed Vehicle Monitoring Testbed - a naval data fusion problem in which real-time issues have also been explored (Lesser et al 1988). Lesser's work builds on his earlier involvement in the Hearsay blackboard system and includes extensions of the blackboard paradigm onto distributed systems. His work on real-time issues harks back to Hearsay's principles of Knowledge Source scheduling; the scheduler should know about each KS the sort of problems that it is able to solve, the quality of solution it is able to generate and the amount of time it will require to produce a solution. The scheduler explicitly constructs a plan for a problem's solution given its resources (in general problem solving processes, i.e. knowledge sources) and the time available to it.

Related to Lesser's "Approximate Processing" ideas is the "Progressive Reasoning" method incorporated in HEXSCON (Wright et al 1986) and in later systems such as (Krijgsman et al 1988). Here a much simpler approach is taken to developing solutions of increasing quality: processing is divided into a number of levels, as events occur the lowest level is run to completion to give an initial reaction, and as time permits further levels are run, in turn, to refine the interpretation.

There are other time-constrained reasoning methods that, for example, pre-filter data and work with what are thought to be the most significant pieces first, and as time permits refine hypotheses by incorporating other data. Such a scheme may, however, suffer from one of the behaviour traits we are keen to avoid, that of Hypothesis Fixation.

The ELINT project at Stanford has been investigating a data fusion and situation assessment problem within the Advanced Architectures Project - a multi-project effort examining parallel processing for AI applications in signal interpretation. See (Rice 1989) for an overview of the entire AAP programme and (Rice 1989b) for results on the ELINT project itself.

In planning, a system is concerned with proposing a series of actions spreading over a long period of time in comparison with that it spends in generating the plan. The question therefore arises as to whether a plan can be generated in time sequence so that the initial actions can be implemented while the plan continues to be extended. The problem, of course, is that planning systems often work with a backtracking algorithm, and any implemented action cannot be backtracked over (in general). The principle is probably a reasonable one for the command and control domain, however, because actions such as aircraft or vessel deployment are reversible, or at least changeable, and there will seldom be a totally inappropriate action. We therefore need planning systems that are not optimising, but satisficing, and that work in this "committable" manner. Clearly the possibility exists to combine this with other time-constrained reasoning mechanisms which will deliver the "best first action" on demand. An idea akin to this committable planning has been explored in heuristic search by (Korf 87).

Lenat et al (1979) identify three "dynamic" abilities and one "static" one which make programs efficient:

Dynamic Self Monitoring and Self-Modification

> The abilities to sense, record and analyse dynamic usage and to use that knowledge to redesign/recompile itself with more appropriate representations, algorithms, data structures (i.e. intelligent learning).

Caching of Computed Results

> Storing the results of frequently requested searches, so they need not be repeated over and over again, i.e. intelligent redundancy.

Expectation Filtering

> Using predictions to ffilter away expected, unsurprising data, thereby freeing up processing time for more productive subtasks, i.e. intelligent focus of attention.

Multiple Levels of Abstraction

> Redundant representation of knowledge at several levels of abstraction can be an economical way of structuring a knowledge base, especially if the program';s tasks are large and the resources available for different tasks vary widely in magnitude, i.e. this is a technique for intelligent knowledge organisation.

They go on to examine mechanisms for providing these abilities.

Issues such as interrupt handling are generally felt to be too low level by researchers to warrant much attention. The basic problems presented to AI systems are much the same as those faced by any designer of real-time software, the indivisibility of certain database operations for example. Competent real-time AI tools will handle these issues anyway, as they will low-level garbage collection.

The tools exist for the exploration of all of the issues in research. Salle and Arzen (1989) discuss features of the three principal toolkits which claim to support real-time AI system development. All of these, quite properly, merely provide a low-level framework without constraining or defining methods for focussing attention, performing macro-scale garbage collection or approximate reasoning.

To take MUSE as an example: at the virtual machine level, instructions are indivisible, so that asynchronous interrupts cannot interfere with database updating instructions, and interrupts can be explicitly blocked by user-defined processes; an agenda supports the scheduling of processes and is used to focus attention within the system, and schedule interrupt handling; an incremental garbage collector runs opportunistically and can be scheduled, objects can be explicitly deleted for space reclamation; two reason-maintenance/multiple worlds systems are supported.

Sloman (1985) is one of the few examples of consideration being given to the deeper issues of real-time operation in Knowledge Based Systems.

2.1.2 Distributed Artificial Intelligence

Several methods have been developed for tackling the issues of problem partitioning, control, communications and consistency within distributed AI systems in a co-ordinated manner, as discussed below. These are commonly termed "metaphors" for their relation to co-operative social organisations.

There has been considerable AI research work into the use of the following metaphors:

- Contract Net Negotiation
- Functionally Accurate Co-operation (FA/C)
- Multistage Negotiation.
- Scientific Community.
- Open Systems.
- Ops Room (as in air defence).
- Co-operation Without Communication

<u>Contract Net Negotiation</u>

The contract net (Smith 1980) uses the model of issuing tenders, inviting bids, and placing contracts with contractors best able to undertake the work, given the constraints of loading and suitability for the task of the contract. Once awarded a contract the contractor may decompose the into a number of sub tasks, to which he acts as a manager, by placing contracts with other contractors. Each top-level task is therefore decomposed into a hierarchy of sub tasks that are dynamically allocated amongst the nodes of the system.

The process of issuing a contract has three major steps:.

(1) Task announcement

A description of the task is sent to those believed capable of completing the task. It comprises a brief description of the work, compliance criteria for the contract, a bid specification, and a `return by' date.

(2) Bidding

Potential contractors first decide whether to bid, and then proceed with preparing a bid which will describe the bidder's capabilities appropriate to the needs of the contract. If the bidder requires information that it does not have, or does not know how to get, the necessary information will be requested by the bidder from the manager of the contract.

(3) Contract award

When all bids have been received, or after the expiry date for responses to the task announcement, they are assessed using application dependent criteria. A contractor will be selected and the contract placed.

Information communicated between nodes is defined using an application dependent, "common internode language", ensuring that all the nodes within the system can communicate with each other. For example, bidders can understand eligibility specifications and managers bid specifications.

There are two questions that appear unanswered from the exposition of the contract net protocol. The first is the problem of what happens when a node receives many tenders and responds with many successful bids. The contract potentially becomes overloaded. Although not discussed, this could be controlled by including within the eligibility specification, possibly implicitly, a request of the form: *Consider this tender only if you can guarantee having the resources available for the duration of the task.*

A second problem, complementing the one above, is that of bid failure. No indication is given to a node that its bid has failed. This makes it difficult for a node to assess the amount of work it may be requested to do. To meet the timeliness of other requests, the node may have to severely under-book its resources.

The very existence of contracts between individual nodes could create difficulty in rapidly refocussing attention of the system. The reason for this is that the complex network of responsibilities set up through the contracts between manager and contractors must be explicitly released. This action may lose significant intermediate results held by contractors that could have been of use in the satisfaction of new system goals. In terms of a mission management role this could represent a very serious bottleneck (for example, the appearance of a threat or other emergency may require an immediate context change to handle the event).

The contract net is highly suited to a distributed processor architecture. The lack of centralised control (although, there are centralised top-level goals to be achieved) also offers increased reliability and the potential for graceful degradation. Further, it offers a more flexible mechanism of control than priority based agenda systems. However, the tendering process is both complex and time consuming and this may be a serious limitation for real-time systems having limited resources. Finally, although an obvious point, it is worth noting that the Contract Net methodology is only of benefit to systems in which there exists more than one node that is able to bid against the various task announcements, or where the dynamics of the system are such that the location of the most appropriate node does not remain constant. These criteria are certainly met by the Naval Command and Control domain but possibly not by the TMD since sensors are fixed.

Application: Distributed Sensing System (DSS)

The contract net metaphor has been supported through the C/NET language used for the simulation of a Distributed Sensing System (DSS) (Smith 1980). The DSS consists of a collection of sensor and processor nodes networked together over a 'relatively large' geographic area. It attempts to construct and maintain a map of vehicle traffic in the area (this application has some resemblance to the Distributed Vehicle Monitoring Testbed described below).

Smith provides an example describing the operation of what he calls the signal task. Here the objective is to associate sensor nodes with manager nodes that perform the signal feature extraction. This is achieved by the manager issuing a task announcement, the eligibility specification of which requires that the node should have a given sensing capability and that the node is located in the same area as the task manager. Potential bidders (sensor nodes) listen to task announcements from each signal manager, and rank the tasks according to the distance the node is from the manager. The node then responds (just before expiry date of the announcement) with a bid to the manager node with the highest ranking task announcement. The manager assesses each bid and allocates the task to the most appropriate sensor node.

Functionally Accurate Co-Operation Systems

Durfee, Lesser and Corkill (1985) describe Functionally Accurate Co-operating Systems by comparing their behaviour to the conventional approach of distributed system design, where an attempt is made to maintain the correctness of processing through task decomposition. These "conventional" systems are termed completely accurate, nearly autonomous (CA/NA) since each node works on complete and correct information and works primarily on local data (The Contract Net metaphor discussed above can be classed as such a system). In contrast, the FA/C approach is for nodes to co-operate in solving a given global goal through the exchange of partial results and hypotheses with other nodes and the use of local information (derived from local sensors/resources) to control its own reasoning in an opportunistic manner.

Systems employing this metaphor summarise their local hypotheses derived from local knowledge and exchange these to develop partial global hypotheses, that is hypotheses that represent the actions, intentions and beliefs of all nodes that are working in parallel on different part of the same problem.

The advantages claimed for such an approach are that it is especially suited for distributed systems in which task decomposition does not map onto distinct processing nodes. That is tasks that cannot complete without significant knowledge of intermediate results from other nodes. An FA/C system also offers increased system robustness since the control in each node is to a great extent *self-directed.*

The Contract Net metaphor was criticised above because of the difficulty of quickly responding to new global goals (change in focus of attention). The inherent local control in the FA/C model allows it to achieve just that, since it only requires a global goal to become locally instantiated to switch the context of problem solving within a node. Intermediate results need not be discarded, as in the contract net approach, but can be immediately applied to the satisfaction of new, more important, goals.

Durfee et al have studied a number of strategies for internode communications within the Distributed Vehicle Monitoring testbed application, discussed below. Strategies investigated for inter-node communication include:

- *Send all*

With this strategy all hypotheses generated by a node are transmitted. Although this ensures a maximum of exchanged hypotheses, it has the disadvantage that a node can easily be distracted by misleading partial results from other nodes, since it uses such results to improve its own reasoning

- *Local completion*

Here only those hypotheses that cannot be improved upon are transmitted. The test of whether a hypothesis can be improved is by examining the queue of instantiated knowledge sources (KSIs). If there exists a KSI that could potentially improve on the hypothesis then the hypothesis is not sent and the system waits until this KSI has run before re-evaluating the situation.

- *First and Last*

Here a node sends the first hypothesis and the last one. The first partial hypothesis is transmitted for predictive purposes (i.e. recipient nodes can use this information to gain a rapid, though weak, understanding of an adjacent node), whilst the last locally complete hypothesis, as described above, is transmitted for integration.

In terms of efficacy, results show that the Locally Complete strategy is the most effective followed by First and Last and finally Send All. Further experiments were performed using local plan generation and exchange of meta-level control information.

- *Local Plan Generation*

A plan is generated consisting of a collection of potential knowledge source instantiations. If this plan intends to improve on the current hypothesis then transmission is delayed, if not the hypothesis is sent immediately. This provides for a more forward looking analysis.

The inclusion of a local plan helps nodes avoid being unnecessarily side-tracked by new hypotheses from other nodes.

- *Meta-Level Communication*

As an extension to plan generation, meta-level communication (passing of control information) to adjacent nodes helps avoid redundant behaviour, by allowing nodes to estimate the effect a hypothesis has upon its neighbour. This is also useful for recognising the occurrence of lost data, since it can anticipate the effects of sending a hypothesis and look for these effects as confirmation of receipt of the data.

Results of providing meta-level communication indicate that solution time becomes near optimal.

Although exchange of meta-level control information delivers a reduction of transmitted hypotheses, the meta-level communication must also consume internode bandwidth. Durfee does not discuss what these demands are.

Application: Distributed Vehicle Monitoring Testbed (DVMT)

The Distributed Vehicle Monitoring Testbed is an application that has been designed to follow the FA/C approach.

It simulates a network of vehicle monitoring nodes, where each node is responsible for a portion of the sensed system and hence each node can concurrently pursue partial tracks. Hypotheses and goals are exchanged between the nodes in order to converge on a complete map of the system.

Each node in the DVMT system is similar in structure to an Hearsay II model with a blackboard with four layers of abstraction. The signal layer for low level analysis of sensor data, the group layer for collections of related signals, the vehicle layer for collection of groups corresponding to a vehicle type, and the pattern layer responsible for collecting specially related vehicle types into formations. Inter-node communication is controlled by specialised knowledge sources that exchange hypotheses and goals among the nodes.

Of all the distributed metaphors the FA/C, and in particular the DVMT application, has undergone a considerable number of experiments (Durfee et al 1985) investigating the performance of the system under varying configurations, including: changing the number of nodes and sensor organisation (i.e. the level of sensor overlap). One important adaptation to the FA/C model used in the implementation of the DVMT, is the use of plans to predict what partial solutions are to be exchanged in the future. This approach significantly improves the performance by making co-operation between nodes more coherent.

Other experiments with the FA/C model have shown a high robustness of the architecture and problem solving method to the occurrence of communication problems.

Multistage Negotiation

Conry, Meyer and Lesser describe the metaphor of Multistage negotiation used for planning in a distributed environment with decentralised control and limited inter agent communication. In practice they attempt to draw together features from both the Contract Net and Functionally Accurate, Co-operative systems.

As with the FA/C model Multistage negotiation is not meant for goal decomposition, but suited to distributed environments with limited communication bandwidth and no single locus of control.

Multistage negotiation provides a means by which an agent can acquire enough knowledge to reason about the impact of local activity on non-local state and modify its behaviour accordingly so as to resolve these subgoal interactions. It also has the ability of being able to deduce when a problem is over-constrained, through the detection of conflicting goals.

The first stage of planning for a node is the generation of plan segments for each of its primary goals. The node then tenders contracts to other agents who will attempt partial satisfaction of the plan alternatives, which they define locally as secondary goals. Each of these agents perform consistency checks of the secondary goals against the agent's own primary goals, and the results are reported back to the node which then ranks the alternative plan segments and transmits revised plans to the appropriate agents.

This strategy is reached by exchanging knowledge about the non-local impact of an agent's proposed local action without requiring the exchange of detailed local state information, and results in solutions which are incrementally constructed to converge on a set of complete local solutions which are globally consistent.

The Scientific Community Metaphor

The scientific community is noted for its ability to solve problems and this provides the prime motivation for this metaphor. Kornfeld and Hewitt (1981) describe the scientific community as possessing the following characteristics:

(1) *Monotonicity*

Scientists publish their results, and once published results remain available to all scientists.

(2) *Commutativity*

Scientists may show interest in a publication before or after it is published.

(3) *Concurrency*

Scientists work concurrently with each other.

(4) *Decentralised Control*

There is no centralised control within the scientific community.

The specialisation of agents within the community is controlled by the introduction of sponsors who 'fund' the activities of the various agents in the community, thereby focusing research. For example, in the real scientific world, research into the development of a new vaccine to counter a new strain of a virus can be accelerated by increasing funding to those groups capable of doing the work.

Kornfeld and Hewitt (1981) use this metaphor in their language called Ether, where a publication is represented as a message and dispatched to all entities, named sprites, that show an interest in receiving the message. This mechanism supports each of the above attributes of the scientific community:

(1) *Monotonicity*

Once a message is sent it remains available forever.

(2) *Commutativity*

If a message is in the interest set of a sprite, the sprite will receive the message irrespective of whether the message was sent before or after the sprite was activated.

(3) *Concurrency*

A single message may be processed concurrently by two sprites, and two different messages can be processed concurrently by a single sprite. That is any method associated with the receipt of a message can run concurrently with any other method.

(4) Decentralised control which allows simultaneous work on multiple hypotheses.

One major disadvantage of actor-based systems in a real time environment is the unbounded nature of response times. That is, the inability to know with any certainty that all responses to a broadcast message have been received (compare this with the contract net protocol approach of defining an expiration time for there turn of bids). The longevity of messages could also be a serious problem for continuous real-time systems.

The examples provided by Kornfeld and Hewitt are rather superficial (in that they have not been applied to serious real-life problems) and the important aspects of proponents, sceptics and adjustment, used to control published theories, are neither examined in detail, nor described through examples.

The advantage of using actor based languages, of which the language Ether is an instance, is that they are inherently parallel. This not only provides for a functional decomposition of an application onto distributed architectures, but also allows for the possible implicit parallelisation at a more local level, the message level.

Open Systems

Open systems represent an extension to Hewitt's earlier work on the scientific community metaphor. Its origins arise from the investigation of the application of information technology to office information systems. Hewitt (Hewitt 1986) sees the office environment as an arena which has to cope with conflicting, inconsistent and partial information.

Characteristics of these systems are:

(1) *Continuous Availability*

(2) *Modularity*

(3) *Extensibility*

In the open systems model all logical deductions are performed within micro theories which are believed to be consistent. If a contradiction occurs a repair is attempted or the theory is split into sub-micro theories which are consistent. Resolving conflict involves negotiation and debate between available and possible overlapping micro theories.

Hewitt introduces the notion of *due process* and describes it as "... the organisational activity of humans (and computers) for generating sound, relevant and reliable information. Due process does not make decisions or take actions it only provides the information necessary for a decision to be made". Hewitt claims that due process is the only kind of system that will work when parts of the organisation do not agree completely and represent different responsibilities. The meaning of the words in rules, policies and goals changes to receive the meaning that an organisation gives to them in due process.

Ops Room Metaphor

In contrast with the previous models, the Ops Room Metaphor (Bell et al 1987) based on the air defence operations room has an explicit control structure with definite delineated responsibilities. This has a defined military hierarchy of responsibility with *Master Controller, Sector Controller, Fighter Allocator* and *Intercept Controller*. Orders are issued down this hierarchy either directed to individuals or groups of individuals. Individuals at any level may also be responsible for maintaining a set of global data on the tote board of the ops room, accessible by all.

CCL used this metaphor to conduct work for RSRE in the development of an AI framework that could be used to experiment with the application of AI to air defence. The primary tool that arose from the work was the languageLOBS (BLackboard ObjectS) where messages (representing orders) could be transmitted between nodes, or broadcast to some subset of nodes. Each node could hold *private* data or make it *public*, visible to all. Other nodes could use demons to the public data to provide asynchronous notification of important changes.

On a distributed real-world problem, the use of global data to represent the contents of the tote board, accessible by all, could have been the source of a serious bottleneck. Fortunately, the structure of the data on the tote board, primarily labels, with a high semantic content(c.f. Smith's application dependent `common internode language' defined in the contract net approach), requires only a limited communications bandwidth to distribute necessary information to each node.

As with the Contract Net, problems are decomposed hierarchically. However,in contrast to the contract net, responsibilities are defined before reasoning commences. This removes flexibility from the application but improves operational performance.

Co-operation without communication

Co-operation without communication must represent the ultimate in distributed reasoning. Although a little outside the scope of the current document the notion of co-operating without communication has many advantages in the defence arena. For example, consider a formation of aircraft either in a defence or attack role. If communication is not allowed between the aircraft as part of necessary radio silence, the intelligent systems aboard each aircraft must attempt to reason and plan on the basis of some global plan and knowledge of the likely behaviour of the agents on the other aircraft.

In their paper Genesereth et al (1985), investigate this problem through a variation of game theory. They discuss two agent systems each of which is capable of making one of two choices. Decision is based on the analysis of a payoff matrix which is assumed to be known by both agents. The benefit of an action by an individual depends not only on the choice made by that individual but also on the choice taken by the other individual. Choosing the best option purely on local information does not necessarily provide the best option (c.f. The prisoner's dilemma). In this simple scenario alone there are 144 distinct interactions between the two agents. Extending it to cover 'real' applications becomes very difficult.

In a later paper Rosenschein and Genesereth (1985) consider the relaxation of these restrictions and introduces communication to setup deals between agents, allowing them to co-operate and coordinate more easily.

2.2 Activities and Resources

2.2.1 Real-time KBS systems

Activities

There is little academic research of interest being done in this area, with the possible exception of Heriot Watt's work on a toolkit for real-time control applications.

Expertise rests with the system builders in the practical concerns relating to systems design, e.g. interrupt handling, garbage collection. CCL's work on MUSE has to be viewed as state of the art in this respect.

Availability of Resource

Most of the recognised KBS houses are are competent in this area: SD Scicon, Ferranti, Software Sciences, Logica, CCL etc.

2.2.2 Distributed AI

Activities

The most well known and long term research on DAI within the UK has been at Essex, initially as the Teamwork project and latterly as IPEM. This work has been capitalised on by others such at British Telecom's research laboratories.

Strathclyde have been exploring DAI in their scheduling work on IKONMAN. The Open University have also been studying multi-agent planning and communication needs in DAI systems. Oxford have been researching DAI in robotics work,including for Data Fusion tasks.

Companies are now beginning to explore DAI seriously. Logica, CCL, BAe, YARD, Phillips Research Labs and STC are all known to be active in the area.

Availability of Resource

In terms of industry resources, Logica, CCL and BAe could justifiably claim that their current IED projects will place them in the forefront of technical competence in the DAI area.

The accessibility of university research, particularly at Essex, may be low because of the military and SDI connections. In this respect, Nottingham, the OU and Strathclyde are better prospects for academic contributions.

3 KNOWLEDGE REPRESENTATION AND MANIPULATION

3.1 State of the Art

3.1.1 Temporal Reasoning

A central characteristic of almost all activities in the envisaged applications is that they take place in a changing world ordered through time. Any advanced AI elements within these applications must be able to express these changes and their attendant temporal information.

Representing Time and Change: The Frame Problem

The remarkable feature of all attempts to devise ways of representing and reasoning about time and change (the attempt to produce an adequate KRL) is the manifestation of a problem - the Frame Problem. The problem arises out of the attempt to reason both rigourously and efficiently about the future. It also appears that the problem does not depend on the underlying method of representing time though the effects may be mitigated by making particular choices of formalism in particular contexts.

The frame problem originally arose in connection with a proposal of McCarthy (McCarthy and Hayes 1969) for representing temporal facts. His idea, which we discuss in more detail in the section on the Situation Calculus, was to represent a fact like 'Jaguar1 is in the hangar' by making explicit the situation it referred to.

Events and actions are represented as transitions between situations. The term result situation,event denotes a new situation resulting from the given event occurring in situation. The behaviour/laws/physics of the world are specified in axioms like

"(x,y,s)(T(s,mobile(x)) Æ T(result(s,go(x,y)),at(x,y))

This can be paraphrased as 'if something is mobile it will be at the location it goes to'.

The bad news is that in such a formulation it is necessary to provide a large number of 'frame axioms' that specify which facts remain unchanged by events and actions. For example going from place to place leaves the colour unchanged. Such an axiom would be expressed in fopc as:

"(x,y,c,s)(T(s,colour(x,c)) Æ T(result(s,go(x,y)),colour (x,c))

Since most things in the world remain unaltered by specific events there will be huge numbers of such axioms.

Actually the Frame Problem is really a number of related problems, as we discuss below. McDermott (1987) distinguishes the following: qualification, ramification and inertia. We shall add a fourth: conservation.

The Qualification Problem

The problem here is to decide the immediate effects of an event given an open-ended list of ways in which it might be modified by context. McDermott points out that this was originally called McCarthy's "potato in the tailpipe" problem. McCarthy's point is made if we imagine an event such as turning the ignition key in a car engine. One normally expects a series of effects to flow from this action. But no matter how careful one is in enumerating what they are or what might cause them not to occur there is always the possibility of an exotic context which produces unexpected effects and which it would be hard to imagine explicitly representing as a possibility before the event happens. For example, a potato in the exhaust pipe may well lead to a series of effects very different than those predicted by the original action. McDermott argues that this is not a problem in temporal reasoning at all, but a problem of the efficient organisation of exceptions to general rules about events. This means that the problem resides in our knowledge about the world - it is epistemological in nature.

The Ramification Problem

This is the problem of predicting all the effects of an action or event. Effects ramify, but how far? For example when one changes the location of an object some things move with it (those things attached to it) other things do not. Again this seems more a matter of the choice of appropriate representations of objects. Any model or representation can only be an approximation of reality. As such it may omit many predictions that will actually happen when an object moves, such as frictional heating.

The Conservation Problem

When an action occurs its effects on the general state of the world may ramify but they are nevertheless limited. In fact most of the world remains unaffected. Deciding how to represent what stays the same is in some sense the obverse problem to ramification.

The Inertia Problem

Here the problem is deciding how long a fact stays true or a process continues. This problem is perhaps the real force behind developments in temporal representation systems and arises because AI has been particularly interested in reasoning about continuous processes.

The Conventional Database Approach

In this approach a knowledge base is taken to be a description or model of the world. As the world changes then the knowledge base undergoes a process of updating. We simulate actions by additions and deletions to the database.

This is essentially a simulation of a change in the world, it represents an effect but it is neither descriptive nor declarative. It is difficult to infer what the change means since there is no explicit semantic description of the meaning of the change. So does deletion of the fact 'Harrier1 is on deck' mean that the aircraft has been destroyed, posted to another carrier or simply taken below deck ?

The philosophy behind this simple database approach is that we take time slices or 'snapshots' of the world. The duration for which such snapshots are deemed to hold is the 'granularity' of the representation and may be an instant, minutes, hours, days etc. Updates occur which represent events and the whole process can be regarded as a series of state transitions.

A further feature of many of these types of conventional representation is that only the latest snapshot is kept.

This form of representation (along with others) is particularly prone to the frame conservation problem. In state based descriptions almost nothing changes from one description of the knowledge/data base to the next. Frame axioms must be provided to specify what does not change. But these axioms can grow exponentially with the number of facts represented in situations.

The Situation Calculus

In the Situation Calculus of McCarthy & Hayes (1969) representational power is enhanced by including an extra argument in predicates. Instead of *status(Harrier1,available)* we write *status(Harrier1,available,s1)*. The third argument indexes the state/snapshot at which the relation holds. Events or updates are regarded as transitions between states. We can write explicit rules which indicate the content of the transitions. We can express declaratively the meaning of transitions.

Moreover, the conservation component of the frame problem can be alleviated by using *Default Reasoning*. The general principle here is that we conclude P in the absence of information to the contrary. A particular form of this is negation as failure - conclude not P if we fail to show P. This form of reasoning can be efficiently implemented.

In such a formulation we have to explicitly state our reasons for believing a state holds both in terms of what has happened and the fact that what has happened cannot have changed the state.

But there are problems with this Situation Calculus (SC) approach as a method of representing change. There is an assumption that our database is always complete. But changes in our database may have missed stages in the progression of events in the world - we may have to fill in the gaps. This type of inference is very difficult to capture in the SC.

Explicit temporal logics

Logical representations of time have typically had to augment the classic First Order Predicate Calculus (FOPC) much beloved of AI workers. This is because simple inferences about time cannot be drawn using FOPC alone. Thus from *Harrier1 is landing* one cannot infer that there will be a time when *Harrier1 will have landed.*

The classic temporal logic approach, excellently reviewed in van Bentham (1982) takes instants and the relation of temporal precedence as primitive. Conventional Temporal Logic adds the modal operators P and F to classical logic. The intuitive meaning of these operators is:

- **PA "A was true at some time in the past"**
- **FA "A will be true at some time in the future"**

The resulting expressivity is quite rich. For example

- **P(F(landed(Harrier1)))**

Some time ago Harrier1 still had not landed

But we cannot refer to moments of time as first order objects. What we need is to be able to refer to 'now' as a time token - treating time as an object to be quantified over. This will allow us to express knowledge like

If the red light is on, fire immediately

But this too is difficult to express with many of the logical facilities that are currently available. What we need is a richer model of time. If we treat temporal objects as first order - then we could choose points, intervals or events or any combination of these as primitive.

In fact much of the current work in temporal logic in AI is directed towards developing systems in which these various temporal elements can be reasoned about.

<u>Representing time as intervals: Allen's calculus</u>

Allen's (1982) system is formulated in classical fopc. He extends fopc to deal with events, actions, beliefs, intention and causality. Allen's ontology of time is a single time line consisting of ordered intervals. Intervals are related by one of 13 mutually exclusive relations.

Examples of three of these relations are shown below

x < y (before)

```
|----x----|

                  |----y----|
```

x meets y

```
|----x----|----y----|
```

x overlaps y

```
|----x----|

      |----y----|
```

Event descriptions imply the holding of relationships (facts) over time intervals. What is central about this way of representing time and change is that reasoning now becomes a matter of satisfying constraints between time intervals.

Allen's system is not a full modal logic. He does not have the modal operators F and P - his logic stays first order. As we shall see this limits expressivity but leads to the possibility of efficient implementations in terms of resolution based theorem provers.

Planning and problem solving in time

Of crucial importance is the ability to efficiently reason with time as well as representing it. Most of the practical work in building systems capable of reasoning about actions and states is in the AI field of planning. Integrating the kinds of representational system described above with activity based planners is very hard. The area of planning is covered in other sections of this report. However, issues especially relevant to temporal reasoning are mentioned here.

Many of the best known planners simply avoided representing time explicitly. In planners such as NOAH (Sacerdoti 1975), and NONLIN (Tate 1976) a problem is described as set of propositions which hold true in the initial state and another set of propositions which we want to be true in the goal state. These planners assume that all actions and propositions are instantaneous. The temporal relations (between two actions) that can be represented are limited to before, after, and in parallel. This limits the planner's ability to talk about goals like:

- **finish sweeping area Alpha and ESM activity at the same time**
- **perform shore bombardment during the period vessels are manoeuvering to offloading zones**

Planners such as DEVISER (Vere 1983) attempt to solve such problems by using the idea of a package. A package can be shared by one or more actions or propositions. It is a window restricting the earliest and latest starting points for certain actions or propositions. It also specifies their duration.

DEVISER can represent:

- **action A takes 5 minutes**
- **event X starts at time T**

But DEVISER has limitations. For example, durations must have absolute values. It can express:

- **to refuel Harrier1 takes 15 minutes**

but not:

- **to refuel Harrier1 takes 10 to 20 minutes**

Firby and McDermott (1988) discuss two attempts to integrate complex temporal representations into planners, namely TMM (Dean 1985) and HTS (Miller 1985). Tsang (1987b) describes TLP, an attempt to integrate Allen's interval calculus into a full blown planning system.

The point with all these systems is that effective planning requires much more than just the concise and powerful representation of temporal information. It requires additional machinery to reason efficiently with this information, to spot possible interactions between actions etc. These issues set the agenda for work in representing and reasoning about temporal facts and events.

3.1.2 Modal Reasoning

While temporal logics are a variety of modal representation, for convenience they were separated out for discussion above because of their special relevance and prominence within command and control domains. In the remainder of this section we discuss modalities of belief, preference, necessity and possibility.

<u>Representing and reasoning about beliefs</u>

There is a growing awareness in artificial intelligence of the importance of reasoning about knowledge and belief, or *epistemic* reasoning. There have been a number of conferences on this topic (Halpern, 1986; Vardi, 1988), and each year a significant number of papers dealing with various aspects of the problem of epistemic reasoning appear in other AI conferences and journals.

One can give many examples of areas in which systems need to be able to reason about their own and other systems' beliefs. One example is multi-agent problem solving systems where agents need to coordinate their actions, and hence need to be able to reason about the likely actions of other agents. The term agent here can be used to refer to a wide class of systems; effectively any knowledge based system that needs to reflect on its own and other systems' states. Such systems may be embodied in complex command and control expert systems such as are typified by the application domains.

The actions an agent may perform depends on its beliefs about the world. Therefore, if agent *a* needs to reason about the actions agent *b* can take, *a* needs to be able to reason about what *b* believes.

Very many forms of important background reasoning are best seen in terms of epistemic reasoning on behalf of a knowledge based system. Truly intelligent systems such as ourselves are able to draw conclusions about the world by introspecting on our own knowledge base, and noticing the absence of certain pieces of information. Consider the reasons a system might have for believing that an aircraft was on course. The system may have no direct evidence to this effect. Rather it draws this inference on the basis of its belief that if it was off course then it would know about it. Moore (1985, 1988) calls this process of a system's reasoning from its own beliefs *autoepistemic reasoning*, and developed an attendant logic to model this kind of activity.

Logics of belief

Most of the work in AI on reasoning about knowledge and belief has used some form of logic as its main knowledge representation. If one chooses to use logic in epistemic reasoning then there are a number of candidate approaches. The foundation for most of them is Hintikka's epistemic logic (1962, 1969).

Epistemic logic has proved to be an important starting point in research on reasoning about knowledge and belief. One of the most important arguments in its favour are the tackling of the issue of solipsis (the relative nature of belief) and the distinction between belief and justified true belief, i.e. knowledge. Secondly, because varieties of epistemic logic are *logics*, arguments in favour of logic advanced by Moore (1984) and Hayes (1977), concentrating on the expressive power (Moore) and the clarity of the model-theoretic semantics (Hayes), are also valid. However, epistemic logics also lead to a number of problems which must also be discussed.

For historical reasons, such logics have traditionally been constructed in an ideal philosophical framework unconstrained by practical programming issues such as tractability. Thus there is a tendency by some of the philosophical adherents of logic to argue that we should only concern ourselves with ideal rational agents, or "superbelievers", and to underplay issues of realism. However, not only are we concerned to build practical systems from the computing point of view, but we also need to simulate the processes of other agents on like assumptions: that other agents are also not superbelievers and able to come to objectively ideal conclusions in finite time. The practical issue of performance of any system cannot be abstracted away from the issue of *omniscience* of the system itself and any other agent in its mental universe.

There are a number of different logic-based approaches to reasoning about knowledge and belief, using modal predicate logic as a base. A modal logic approach takes a unary belief operator, *B* , to represent an agent's beliefs. Thus if F is a well-formed sentence, *B* F is the representation that the agent believes F, and is also a well-formed sentence. By adding subscripted belief operators, it is possible to represent the beliefs of many agents in the logic.

In systems such as KD45 (Chellas 1980) the following propositions can then be shown to be valid:

(a) all axioms of classical predicate calculus

(b) (BF & B(F Æ X)) Æ BX (consequential closure)

(c) ¬BF Æ B¬BF(negative introspection)

(d) BF Æ BBF(positive introspection)

(e) BF Æ ¬B¬F (consistency)

Problems with logics of belief

There are a number of potential pitfalls in using any representation language to prescribe a system of belief. Firstly, systems must be solipsistic to a degree. Predicting human agents' behaviour on the basis of assumed belief requires that there is a recognition that all beliefs are *beliefs of agents*, and that there is no all-seeing, all-knowing 'god's-eye' view of the system. This applies equally well to distributed co-operative artificial intelligence.

The problem of *logical omniscience* is more a symptom cluster than a single issue. We can break down what is essentially a qualitative performance comparison with (human) belief into four areas, using Fagin and Halpern's (1985) categories. These are *lack of awareness, resource-boundedness, lack of inference rules and limited focus of attention.* These problems themselves are symptoms of the nature of ideal rational performance epitomised by logic, contrasted with tractable machine performance, or, for cognitive purposes, even more limited human performance. For example, humans tend to lack even basic inference rules (such as modus ponens), whilst we may be able to assume that all machine agents were designed so as to behave with all axioms of predicate logic at least. However, when computing a set of beliefs that a human operator is believed to hold, the system should account for human failings and not naively assume good logical abilities of humans.

The issues of resource-boundedness and focus are clearly associated with the processing procedure and heuristics driving the logical deduction mechanism. Description of what constitutes a resource- or focus-boundary is dependent on the underlying mechanism. In order to perform tractable and useful applications of rules a 'lazy', as opposed to 'zealous' evaluation method is required, evaluating rules when necessary, rather than causing an exponential computation explosion. In order to do this successfully, rules must be prioritised and processing focussed. Relevance heuristics, for example, cause prioritisation of the application of certain rules and in the process, necessarily narrow the 'focus of attention' of the system. Thus it is better to describe these boundaries as functions of underlying processing rather than conceptualise them as physical limitations such as 'bandwidth' or 'memory size'.

A lazy-evaluated system, however, certainly cannot guarantee consistency and logical closure, precisely because to do so would mean application of rules zealously. Propositions (b) through to (e) of KD45 above would no longer hold from an omniscient observer viewpoint, and if we were to ask questions of the system which did not cause an inconsistency to be resolved, inconsistencies in belief could be detected. This is akin to the observation that humans may well hold contradictory beliefs, holding to them independently, but when confronted with the contradiction may attempt to reconcile them. Introspective consistency is only resolved with lazy evaluation. You may believe you believe x, but it is not necessary to generate an infinite chain of "I believe I believe I believe... x" (and it is difficult to justify in order to solve a practical problem).

There are a number of proposals in the AI literature for avoiding the problems of logical omniscience, at least in epistemic logic. Konolige's *deduction model of belief* (1986) restricts inference rules for generating new beliefs from old ones, confining them to a 'deduction structure'. Reichgelt (1989) points out that inference rules can be formulated in terms of reified epistemic logic and can deal with resource limitations on a *cost basis*, where the 'cost' of a belief is attached to the belief with a two-place predicate *D* and a one-place predicate *bel* corresponding to the costs and the belief set of sentences respectively, for example *bel([F]) & D(0,[F])* where *F* represents a well-formed sentence, and *[F]* the name of that sentence. Imposing an upper limit, *i*, on the depth to which beliefs may be explored is seen as one way of restricting the boundaries of resource limitations. As Reichgelt comments, such a system can deal with some aspects of logical omniscience: lack of awareness and inference rules (syntactic restrictions), and, numerically, with resource boundedness (and focus, as a corollary). However, on the face of it, such a solution is somewhat counter-intuitive to the notion of a process of conscious reflection in human agents, and, costing systems should be secondary to the active focus solution embodied by lazy evaluation: in other words, use the latter heuristic only when processing seems to be getting nowhere.

These problems have led some researchers to propose very different methods of representing and reasoning with belief.

Alternative methods of representing beliefs

The exist a number of very different approaches to representing belief and other modalities. Thus Rapaport (1986) investigates the representation of reasoning about belief in the *semantic network* formalism, SNePS. Moore and Hendrix (1979) also look at the use of semantic networks for the representation of belief, whilst Ballim and Wilks (1989) use a *procedural analysis of belief* in terms of processing spaces.

Ballim and Wilks argue that the logical method is weak in inference. There is no logical implication between the truth of a statement and whether an agent believes it, or of one agent's belief about one intension and another agent's belief about a corresponding intension, even if they are to the same object. These non-entailments are not remedied by recourse to logic, instead, argue Ballim and Wilks, it is necessary to use *extra-logical* heuristic methods. Their core proposal is thus for a *procedural approach* using certain default rules on beliefs partitioned into 'viewpoints'.

They deal at length with the question of how computation should proceed in evaluating such belief nestings as "the system's view of A's view of B's view of... Z". A distinction they make is between 'topics' (intensional objects) and 'agents' (intensional objects with beliefs). Due to the solipsis requirement, the primary agent is, of course, the system. The system's beliefs form a tree structure with agents and their atomic beliefs at the nodes. This data structure is defined, naturally, by the operations performable on it. Foremost among these is the constructor termed the *default ascription heuristic* which ascribes beliefs to agents in the system's universe. This provides a mechanism for assuming beliefs of other agents unless evidence is found to the contrary. It inherits beliefs down the tree unless contradicted (as and when required by lazy evaluation). This form of inheritance is similar to, but not identical with, categorical inheritance.

A secondary heuristic, formally stated, is the truth assumption: that is, assume utterances are true unless we have reason to do otherwise. Default ascription is qualified by a notion of competency: how competent are agents to hold beliefs (as believed by another agent). Competency is hierarchically ordered into categories of *competent with respect to...*, *capable of evaluating...*, and, most strongly *knows how....* Lecture situations are said to be subsumed under the weakest category: assume that the speaker is speaking from knowledge. The utterances are dealt with a minimal strategy: assume true.

The system operates, as mentioned above, on a lazy evaluation basis. In order to decide what to evaluate, a simple notion of 'relevance' is introduced. This forms the key heuristic in process control, and, compared to the evaluation process itself, needs to be relatively computationally trivial. Ballim and Wilks' suggestion is for a simple 'mentioned' heuristic: if an agent is mentioned explicitly by an incoming utterance, then process that utterance relative to that agent. They admit that their current program heuristics will only account for a limited proportion of the full relevance capability a complete environment-generating system should have.

One of Ballim and Wilks' key arguments in favour of their approach is that environments should be considered at the *formalisation level*. The crucial distinction between the authors' work and modal epistemic logic is that in the latter, such partitioning is at an *implementation level*, so as to leave heuristics formally stated in logic and explicit (and slower). On the other hand, all other heuristics are subordinate to partitioning. Relevance heuristics and meta-beliefs must be handled through environments. The hard-wiring of heuristic search by explicit partitioning at the formal level is argued on a practical engineering criterion, not a psychological or logical one.

Representing and reasoning about necessity and possibility

Turner (1984) gives a clear description of modal logic and the operators M and L, the possibility and necessity operators respectively. In this logic MF represents "it is possible that F is true", and LF represents "it is necessary that F is true" (ie. F is always true). As any modal logic, the standard semantics are those of possible worlds. Formally, a *modal frame* is a Kripke structure $[W,D,R,F]$ where W, D, and R are defined as for epistemic logics (ie. W represents the set of all possible worlds, R the accessibility relation between them, and D the set of individuals in the worlds). The function F assigns interpretations to expressions.

Moore (1985) develops a modal logic of knowledge and action using this approach. The Kripke model above is converted into Hintikka's epistemic (knowledge) logic semantics by introducing a relation K, similar to R, such that $K(a,w1,w2)$ means that for agent a, worlds $w1$ and $w2$ are indistinguishable from one another, and thus accessible possible worlds. K is taken to be reflexive and transitive on $w1$ and $w2$, and thus the underlying logic is S4. Statements about knowledge are only true relative to a possible world, so Moore introduces a two-place predicate $T(w,A)$ which means that the formula A is true in the world w, and introducing the actual world $w0$. Thus for all formulas A, the truth of A (called TRUE(A) by Moore) is $T(w0,A)$. This notation, however, leaves us with an omnisciently observed universe, where the solipsis requirement is ignored. Dropping this notion of 'knowledge' and leaving us with 'belief' returns us to the model of solipsistic epistemic agents who hold retractable and assertable beliefs about the world, but with no *a posteriori* knowledge.

Representing and reasoning with other modalities: commitment, preferences and permissions

Cohen and Levesque (1987) are concerned with the *rational balance* needed among beliefs, goals, plans, intentions, commitments, and actions of autonomous agents. In short, they attempt to provide a theory of *rational action* in modal logic. Such a theory may provide a realistic basis for generative planners. This 'rational balance' is essentially a heuristic description of how an agent's beliefs, goals, and intentions should be related to its actions. Traditional approaches to planning intentions are operational ones: intentions are merely the contents of the plans. This lacks precision, especially in dynamic domains when goals may need adding, extending or abandoning.

Cohen quotes Bratman (1986) who argues that intentions play three functional roles: *(a) intentions cause the agent to determine a plan, (b) intentions provide a 'screen of admissibility for adopting other intentions*, and *(c) agents monitor the success of their plans*. Further, intending has additional properties: that (i) the intention is *possible*, that (ii) the agent *believes he will perform the crucial action*, (iii) *under certain conditions*, and (iv) that *side-effects are not intended* unless specifically required. The last point is that side-effects may be *chosen*, but not *intended*. Their agents are characterised as "not striving for the impossible, and eventually forgoing the contingent" and committed to their goals to varying degrees.

The model of belief chosen as an example by Cohen and Levesque is KD45 and suffers from the problems of logical omniscience detailed above. Secondly, their definition of competence is defined with a KNOW operator which maps intensions to the real world, challenging the solipsistic criterion. They quite reasonably assume that the agent is conscious of its own actions and rule out accidental/unknowing execution of primitive actions. The system of actions in the world are restricted to an essentially serial *course of events*.

Cohen and Levesque's system is not sufficient in itself for plan generation. "The best we can do, and that is not too far off, is to say that the agent has an action expression in mind. That is, we would want to say that eventually, the agent forms a plan..." The problem is that the logic does not permit quantification across action expressions.

What is 'rational' in terms of one individual is not necessarily rational for collective agents. A key aspect of *conflict resolution* is the notion of individual rationality *relative to the collective rationality* of the distributed system.

Galliers' (1989) work concerns itself with conflict between agents, adopting a strategy of flexible conflict resolution, rather than treating conflict as something to be ignored or avoided. "Conflict is considered to be a positive force in the maintenance and evolution of cooperative multi-agent systems, because its expression and consequent potential resolution or management makes possible a flexibility in dealing with unexpected events... In contrast, existing systems are rigid and constrained by imposed benevolence." Galliers attempts to apply social psychological concepts of conflict resolution to distributed AI.

Conflict resolution is to `change someone's mind', to change an agent's goals according to prioritisation decisions, which each agent must share. The concept of *preferences*, maximum satisfaction of goals, implies ensuring the greatest consistency with the *values* the agent holds. We can see how incorporation of such an approach is vital in the types of application envisaged in the problem domains. Thus getting pilots safely down from the sky might have the highest priority value, thus aircraft recognised as being 'in trouble' should be prioritised. Resolving a conflict between two planes wishing to use the same flight path may be done on the basis of which has less fuel, structural stability, and alternative choices of action.

Galliers' approach is to extend Cohen and Levesque's notation to conflict resolution. A preference describes the relationship between a belief about a pair of propositions, and a goal. They specify which proposition to retain, and which to reject, should a particular situation arise. An agent preferring p to q is defined as the agent having a belief that in a situation where she believed that either p or q would be true in the future, she would choose p. Preferring action p to q also means that if the agent believes p has a consequent r, then she also prefers r to q.

Janet Bruten, and colleagues at Hewlett Packard (Bruten, 1989; Kidd, 1989, Allport and Kidd, 1989) use a many-sorted *deontic* logic which is intended to capture the notion of agents commanding, forbidding or allowing other agents to perform certain actions. Each agent has a *permission structure* which prescribes which actions are permissible or impermissible, and is persistent over actions, although permissions may be altered by the act of performing an action and changing the state of information about the world. Obligations, on the other hand, may be incurred by agents at any time, with the restriction of only one obligation per agent at any one time. The obligation interrupts the agent and suspends the permission structure until the obliged action has been fulfilled, or when another agent suspends the obligation. When an obligation has been carried out, new permissions may be altered again. The use of a deontic logic, claims Bruten, allows them to produce a *readable specification with a clear conceptual mapping* to their source knowledge, and to characterise and distinguish between *structural information* about the domain and *behavioural information* (ie. heuristics) about the performance of the task.

Bruten et al use the modal notation *[a]*D to represent "if the action *a* is performed and terminates, then D, the resulting context, holds in the resulting state". *[a]*D and D are both formulae in the logic, so any number of modalities and modal operators may be incorporated in D. A useful construction is D Æ [a]G which acts conditionally on D. They then extend the action logic to include deontic modalities, *must* and *per* (permit). Thus D Æ *[a] must(b)* means "if D holds at the moment, and *a* is performed and terminated, then force the obligation to perform *b*".

Such a logic requires incorporation within an epistemic modality for multiple agent interaction, or in Ballim and Wilks' procedural approach. The asolipsistic confusion between *states of information about the world* and *states of the actual world* can be avoided. In terms of modelling *human* agency, permission structures are no help, however, in placing explicit formal restrictions on legal operations of system users (interactors ?) they could provide a powerful approach. Human agents requesting an action of the system may well be considered as demanding the system fulfill an obligation.

How to choose a representational approach to modalities?

The question one wishes to ask at a practical level is whether the various proposals for representing modalities display advantages over one another.

In this context we will consider the following requirements on our representational systems; scope of expression, naturalness of expression, upwards compatibility, and computational tractability.

Scope of Expression

This criteria relates to the range of modal information that can be represented in any calculus. One can assess different systems in terms of what they could and could not express. What one needs to do in the context of this project is isolate those critical aspects of modal reasoning that present themselves in the various application areas.

Naturalness of Expression

This concerns whether or not the notation is easy to understand and whether it embodies the modal relation succinctly and concisely. This is somewhat subjective since some AI practitioners will go to extraordinary lengths to claim that the most opaque constructs are perfectly readable.

Thus if we are reasoning about time then the English statement below is extremely expressive, succinct and clear. Since our machines cannot reason in natural language we would like our formal language to have some of the virtues of clarity that the English sentence below has.

We know that sometime between now and three hours ago the aircraft crashed. What is in the past now was a future event three hours ago. For a system to conclude that a crash has occurred in the interval it requires a representation something like:

P(crashed(Aircraft1)) & AT(3-hours-ago)(F(crashed(Aircraft1)))

Three hours ago Aircraft1 took off, its wreckage has just been sighted.

It is a moot point how transparent this representation is - whether or not for example we should provide translation mechanisms between our KRLs and their users.

Upwards Compatible

This criterion has to do with ease of integrating one modal calculus with some other system for representing other types of information. For example, temporal and epistemic systems need to be capable of integration. This would allow us to represent knowledge of the form *Controller of radar1 knows that aircraft1 will be routed to EMA.*

Computational Tractability

Finally we should mention the problem of implementation. The inclusion of modal operators in any logical approach has an important effect on the computational implementation of such logics. In efficient first order theorem provers the formulas P and $\neg$Q may be *resolved* against each other only if P and Q *unify*.

The intuitive idea is that it is impossible for a formula to be both true and false. But to establish this in, say, the temporal modality we have to check if the contradiction arose at the same time index. Thus in addition to checking if the formulas unify we also have to check that they unify at the same time indices. This is an additional and potentially very expensive extra check. All modal theorem provers incur this type of penality - it arises whenever unification has to apply across possible worlds and/or time indices.

For non-logical approaches the issue of computational tractability is again crucial. Thus in any semantic network based approach we would want to have benchmarks on the storage, modification and search of such data structures.

3.1.3 Model Based or Deep Reasoning

Most current KBS systems are based on simple heuristics. Thus a fault detection system will consist of lots of associative rules that relate various states of sensors to underlying fault states. But they have no idea of how the states might cause the symptoms (or indeed whether they are side effects). This type of knowledge is called 'shallow,' 'surface' or 'compiled' knowledge.

Heuristically based systems have implicit weaknesses. In such systems each case has to be explicitly pre-specified. There may be so many cases as to make it impractical to enumerate every one. And having to explicitly state all the preconditions for each solution is an error-prone process.

The resulting systems tend to be frail and inflexible: they will fail when presented with circumstances even slightly different from those anticipated by their creators. Shallow systems are often difficult to maintain since what is conceptually a single piece of knowledge may be distributed across several objects in the system. Moreover, the explanations of such systems tend to be simple recaps of the chains of inference that led to a conclusion.

Types of expertise used in reasoning

Many areas of expertise do rely heavily on shallow associative knowledge. What happens when a rare or difficult case arises which doesn't fit any of their existing heuristics ?

Experts rely on other types of knowledge. We can distinguish at least; strategic, structural and causal knowledge.

Strategic knowledge is to do with how to solve the types of problems an expert encounters. For example the generalised hypothetico-deductive model problem solving contains strategic knowledge such as "First establish some feasible hypotheses and then eliminate the least likely of these".

Structural knowledge reflects the fact that domains can be organised into hierarchies and taxonomies. Symptoms, tests, remedial actions, etc. can all have such structure. This makes it easier to relate knowledge and view it at different levels of abstraction.

Causal knowledge describes a model that allows one to describe the functioning of a system in terms of underlying mechanisms. It is causal knowledge that is at the heart of deep reasoning.

The advantages of deep knowledge

Deep knowledge can be used to explain and justify the conclusions of empirical knowledge. It can provide a useful check on the shallow conclusions.

In suggesting a course of action it may not be possible to deal with the main cause of a threat, but it might be possible to deal with an intermediate link thereby relieving some, or all, of the immediate danger.

In rarely encountered cases there may be no solution suggested by the associative heuristics. The system will need to reason from first principles about the possible causes.

Where two contexts both suggest the same hypothesis, causal reasoning can be used to decide whether they provide independent support for the hypothesis, or whether one of the situations is a consequence of the other.

Building models

But how do we represent this deeper, causal knowledge? The commonest approach is to build a model of the real system. This model attempts to describe the real mechanisms as instances of more abstract ones. Causal links may be represented and reasoned about explicitly, or the behaviour of the system may be simulated and reasoned about from first principles. Two types of models can be distinguished: static models and dynamic models.

Static model representations are used to support reasoning about relatively constant aspects of systems, such as the interconnection of components and known causal links. They are especially useful for diagnostic and failure isolation applications.

Dynamic models can support reasoning about time-varying characteristics of systems. The models often use reasoning from first principles to infer the behaviour of a mechanism from its structure and initial states.

Using these models to support reasoning provides a 'safety net' for the shallower empirical knowledge. If the empirical knowledge fails then the inference mechanism can apply general reasoning principles to the model representation. Consequently, such systems show higher performance at the periphery of their knowledge, and are capable of handling problems which were not anticipated. Because of the explicit model representation and close mapping to the real physical system, it is normally easier to verify the completeness of model based systems. These systems are also capable of generating better explanations, since reasoning steps which are implicit in shallow models can be elucidated.

However, model based systems do tend to be comparatively slow and involve more complex reasoning than shallow systems, since a sophisticated control structure is usually required. From what has been said it should be clear that there is no hard and fast boundary between shallow and deep knowledge, merely a continuum of depth between the two.

Static Models

There are several aspects to the static representation of models. For a start, there is a description of the types of components involved in the model. This often takes the form of a hierarchical taxonomy. We can also represent the topological structure of the model. This involves a static representation of the interconnections between devices, the devices being particular instances of those described in the class hierarchy. The connections may also represent potential causal links between components. From this knowledge of the interconnections and the behaviourial description of each component, the procedural, rule-based, part of a KBS can reason about chains of cause and effect.

In general the effects on connected components caused by the change or failure of a certain component can be studied. Or, given a set of symptoms, the system can attempt to reason about the internal states giving rise to the observed symptoms.

The specific benefits of using static models are clarity and communicability of the representation. The model structure is explicit, not embedded in rules. Clarity also implies simplified maintenance. Moreover, since more of the knowledge is explicitly represented in the frames of the static model, it is often possible to simplify the rules to cover fewer more general principles, rather than many specific cases.

Dynamic Models

Dynamic deep models attempt to animate the causal relationships embodied in a system. By propagating quantitative or qualitative changes around a model it is possible to reason about the behaviour of a system. The model usually consists of a detailed representation of the structure of the system, the causal links being derived from the behaviourial interactions between the components.

Dynamic models are especially useful for exploring hypotheses which cannot be tested on the real system. This might occur where the system is too dangerous to test, there is no measuring equipment, the system has failed, it is in production use, etc.

For many years scientists and engineers have attempted to come up with fully deterministic numeric/quantitative models of systems. Alas, it is not always possible to build a complete and entirely correct mathematical model and even if successful, the computation needed to solve it may prove to be prohibitively expensive. Instead of numeric models with variables representing real number quantities, we can use qualitative models.

Qualitative reasoning attempts to provide representations and inference methods that are simpler and more efficient than quantitative models and yet are powerful enough to perform some important reasoning about physical systems.

Unlike quantitative variables, qualitative variables only take on a few discrete values. The variables can be used to indicate the value of a parameter relative to some nominal value, for example 'high', 'normal' and 'low' or 'positive', 'zero' and 'negative', 'decreasing' and 'steady'.

With qualitative reasoning it is not always necessary to have a complete model or complete set of data in order to reason about the behaviour of a physical system. The model only needs sufficient detail to support qualitative reasoning, and similarly only a qualitative description of the initial state of the physical system is required. The reasoning is then used to derive possible subsequent states, given the initial state and information such as the topographical description and the behaviour of individual components or groups of components.

However, qualitative prediction like this won't in general give exact predictions: there may be ambiguities in the conclusions because of the incompleteness of the model.

A dynamic model improves the inferencing capability of a knowledge based system. Using the model it is possible to determine information and estimates for values which would otherwise be unavailable. It can also support reasoning about temporal aspects of the modelled system.

There are drawbacks. A dynamic model entails a large overhead in terms of the speed with which it can reach conclusions compared to shallow-based systems. Trying to come up with a set of general principles with which we can qualitatively model our own common sense, knowledge about physical phenomena, such as forces, fluids, motion, time and so on is a substantial research problem.

3.1.4 Reason Maintenance Systems

There are a number of issues which can be treated under the heading of reason maintenance: data dependencies, hypothetical reasoning, non-monotonic reasoning, and belief revision strategies. Work in the field involves inference recording devices or extensions to formal logic.

<u>Data dependency</u>

Data dependency techniques involve the indication that one fact in a knowledge base is dependent on another for its state or its presence in the knowledge base. One obvious example here is the use of a network to link propositions. With new information coming into the system, new inferences being made and old conclusions appearing unacceptable, these links need to be altered. This is a task for a reason maintenance system (RMS), which is conceived as a separate module from a Problem Solver, which receives the new information and performs the inferences which are subsequently recorded. However, the problem solving in some domains (data fusion, for example) can involve the setting up and manipulation of links between representations of domain objects, rather than between propositions about the domain objects. In such circumstances a separate RMS could be superflous.

Hypothetical reasoning

Hypothetical reasoning ("What if?" queries) is another target area for RMSs. By examining the links which can be built up between propostions one can inspect the different facts which can be derived under different assumptions. Recording the assumptions under which items are true can effect a partioning of a knowledge base, thereby allowing a measure of reasoning from conflicting information or beliefs. Alternatively, one can query which assumptions would need to be true in order for certain facts to be asserted. A formal treatment of the foundations of one RMS, the Assumption-based Truth Maintenance System (ATMS) (de Kleer 1986), has highlighted that such hypothetical or abductive reasoning is its chief utility.

Non-monotonic reasoning

Non-monotonic reasoning can be described crudely as reasoning where more information refines ones beliefs. This is characteristic of situations where one makes inferences whose basis is the absence of other information. In RMSs non-monotonic inferences occur when, instead of merely recording the dependencies between inferred data (which is a standard proof method of classical logics), one uses the state of that record to define new connectives, operators or inference rules. For example, one can introduce statement forms to the effect that proposition P can be inferred from Q if there is no proof of R.

The Justification-based Truth Maintenance Sytem of (JTMS) of Doyle (1979) records the propositions immediately involved in deriving a proposition together with a list of propositions which had to be "out" in order for that proposition to be "in". De Kleer's Assumption-based Truth Maintenance System (ATMS) also records the items immediately involved in an inference but is mainly concerned with noting which unproved items (premisses or assumptions) an item depended on; it does not keep an out list in the manner of Doyle. Martins and Shapiro's "Semantic Network processing system with Belief Revision" (SNeBR) (1988) conceived earlier than De Kleer's ATMS but published later, combines the focus on assumptions (called an "origin set") with an analogue of an out list (called a "restriction set"). They have a more sophisticated understanding of standard logic techniques than either de Kleer or Doyle but the system has not generated much secondary literature nor been used in applications. There are a number of reasons for this. Publication of the full system is quite recent (1988) and it requires the definition of a special-purpose logic.

Stripped down to their core concepts the JTMS involves a non-monotonic reasoning strategy; the ATMS of de Kleer doesn't. Hence the complaint of Reinfrank et. al (1989) that Reiter & de Kleer's formal foundation (1987) does not address the issue of "non-monotonic justifications". Developments to de Kleer's ATMS have been continuing. A backward chaining ATMS and a "first order ATMS" (the node label for a proposition accompanied by a list of variable bindings) have come from Ginsberg (1989), for example. Some of this work involves accepting and extending the pre-suppositions of de Kleer on how the ATMS is to be used on certain problems, rather than disentangling the basic proof recording machinery from its original accompanying illustrations. de Kleer himself has added a "massively parallel ATMS", an ATMS which allows negated assumptions, and a comparative study of ATMS and constraint satisfaction techniques (Dixon and de Kleer 1988, de Kleer 1988, de Kleer 1989).

The alleged benefits of reason maintenance systems have come under investigation. No-one is interested in the use of RMSs to provide explanations since a data dependency record seems scarcely an improvement on a rule trace. RMSs can be used to cache intermediate results. This depends for its utility on the characteristics of the problem solving techniques one is using to derive the results in the first place. If a search algorithm gets it right first time, for example, there would be no need to store intermediate results to avoid later recomputation. Even in the absence of this ideal case, the trade off between time spent re-computing and time (and space) spent recording needs to be gauged relative to an application. An analogous observation can be made on the use of an RMS as a consistency device. Priest (1989) has attempted to interest AI researchers in "paraconsistent" logics, in which the licence to infer any proposition from an inconsistency is revoked. One such - the relevance logic of Anderson and Belnap (1975) - influenced the design of the belief revision system SNeBR. The chief distinguishing factor is the stricture against proving any proposition whatsoever from an inconsistency: only if the proposition to be inferred played a relevant role in the derivation of the inconsistency does the proof go through. Given the prominence of refutation-based methods and the fact that an initially inconsistent knowledge base could be used to prove anything, one might have expected this area to be more studied. However, one finds that in practice either the relevance criteria are observed or that an inconsistent knowledge base, prior to evaluation of a query, simply cannot be expressed.

There are a number of systems or extensions to standard logical techniques which can exhibit the phenomenon of non-monotonicity in the strict sense. (With sets of propositions of a language, A is a subset of B but the theorems provable from A are not a subset of those provable from B). The two favourite formal developments are circumscription and default logic. Articles on each technique regularly appear in the main research journal (Artificial Intelligence), with the balance in favour of the former.

Put simply, circumscription involves the addition of axioms or axiom schemas to a set of sentences (a knowledge base) in a manner which limits the interpretation of the constituent sentences. In this way it permits something analogous to closed world reasoning. The question of how best to introduce and exploit these extra axioms, and how to keep track of the inferences made from them, is not considered part of the technique itself.

Circumscription comes in three main forms: predicate, formula and domain circumscription. However, consideration of the use of circumscription in problematic cases has led to notions such as pointwise, prioritised and protected circumscription, and an extension to cover the use of the equality predicate has been proposed. The additions to the basic idea to make it applicable to particular cases reduce its formal simplicity and weaken the intention of providing a principled treatment of non-monotonic inference.

Automated reasoners for circumscription have started to appear. However, the earlier work by Przymusinski (1989) providing an algorithm for computing circumscription relies on a special purpose resolution technique and the theorem prover of Ginsberg (1989) is a simple query evaluator which still leaves much room for development.

Default logic, introduced by Reiter (1987), is a simpler but non-classical technique which introduces new rules of inference allowing the derivation of a conclusion provided certain consistency conditions aren't violated. A non-monotonic logic developed by Doyle & McDermott (1980) extended the logical vocabulary with a consistency or non-provability operator which could be applied to sentences containing domain-specific non-logical vocabulary to state default knowledge. By contrast, default logic seems to require domain-specific default knowledge to be introduced as inference rules. This is not really consonant with the idea of a logic as providing tools (sound rules of inference among them) for assessing the validity of arguments independent of subject matter. In practice, the distinction between using a default operator and a default rule of inference can be blurred.

More immediate difficulties for default logic are the problem of interacting defaults and the utilisation of consistency tests in the definition of its inference rules. Interacting defaults, such as might be used to translate "Typical As are not Cs, typical As are Bs, typical Bs are Cs", can lead to inconsistencies. There are similarities here to the failure of transitivity in inferences involving counterfactual conditionals. To avoid this problem requires introducing new or more complex definitions or the use of proof recording devices to prohibit default inferences leading to the unintuitive conclusions. The latter will be needed in an applied default logic in any case, to retract information inferred by default. The second problem arises because questions about consistency or provability are not in general fully decidable. More experimentation is required to discover tractable cases, and to decide what resources to devote to consistency tests in a practical system.

A significant amount of work goes into arguing for the representational capacity of circumscription and default logic, developing the model theory and undertaking comparative studies. Applications where the developers claim to be employing circumscription or default logic are hard to find. Etherington (1988) has shown the utility of such work as a means of formally examining the properties of other representations such as inheritance networks. Investigations of commonsense reasoning about inheritance form a favourite topic of more recent conferences.

<u>Belief revision</u>

Belief revision comprises both a technical and a practical issue. RMSs are largely concerned with identifying inconsistencies and ensuring that the network of linked propositions achieves some consistent state in the face of new information. Their focus tends to be on the computational efficiency of constraint propagation, search and backtracking techniques. The practical question of the best strategies to adopt when new information makes a revision of beliefs necessary has given rise to a number of theories. Examples of such revision strategies are maximal coherence of beliefs and minimal change to them in the face of conflicting information.

Alternatively, one can class some formulas as protected, or rank sentences in order of revisability. In this area the work of Gardenfors (1988) on rationality postulates is most often quoted. These theories are based around modal logics of belief and typically involve finding interesting mappings from one possible (belief) world to another. Axiomatisation of the characteristics of belief using possible world semantics is also the province of Moore's Autoepistemic Logic (1988). Logics of belief revision extend the basic triple of agent, formula and world to include a specifiction of time. A related, though much older technique, which aims to find maximal consistent sets of sentences (beliefs) in the face of conflicting information is Rescher's (1975) plausibility theory.

Range of Work

Some research work involves hybrid systems containing elements from several sources, for example, the use of default logic to specify and validate a hybrid JTMS/ATMS system. Other work aims to demonstrate the relationship between the various approaches. For example, the "in" and "out" lists in a JTMS proof record can be translated to the pre-requisites and justifications of a default inference rule. Alternatively, default reasoning can be reformulated as closed world reasoning with circumscribed abnormality predicates. There is a general desire to combine the practical utility of the existing RMS proof-recording approach with the perceived logical respectability of the more formal developments.

There are number of topics in the area of reason maintenance which require work. One is a computationally feasible system for circumscriptive or default reasoning, which is needed before the utility of such formalisms can be assessed in applications. Another is the utilisation of data-dependency techniques in real-time and continuous reasoning domains where convergence to a solution, characteristic of the usual RMS examples of consistent labelling or constraint satisfaction problems, is not an applicable notion. This would highlight any necessary trade-offs between storing dependency information and saving on space and time. It would also allow one to assess whether recording data dependencies in those circumstances is counter-productive: perhaps the faster an inference system the more a reason maintenance system has to record.

With an RMS there may be a large overhead involved in working out the required updates to a proof record when a new inference is reported. It may be better, for example, to use a blackboard architecture for a problem solving module, keeping conflicting conclusions in separate knowledge sources, than to have a record of the different environments under which globally accessible facts are true. However, one finds little comparative work of this sort being done with large applications. No studies have appeared concerning the utility of a distinction between inferred data whose origins need to be recorded, and inferred data which can be simply stored. Little has been done on showing whether it is worth bothering with a reason maintenance system in a language with side effects. The issue of how best to adapt the inference recording techniques to a distributed system is unexplored. Unfortunately, such practical issues are considered to be, in de Kleer's terminology, a matter of the interface between a Problem Solver and an RMS module.

3.1.5 Uncertainty

The handling of uncertainty in AI can be split into numerical and non-numerical treatments. The numerical treatments tend to be have a more dubious intellectual basis which is covered by the pragmatic success of applications using them; the non-numeric techniques can be better models of problem solving but it is harder to make them do real work. Research work tends to assess the value of one particular technique for handling all uncertainty. Applications work tends to be more flexible in the variety of methods it uses or the different forms of "uncertainty" it identifies. A frequent distinction is that between vagueness or imprecision due to measurement or qualitative description of a scale reading and uncertainty due to less than whole-hearted belief in a statement. The further division of the concept to include intrinsically random systems is not common. There are no treatments which use different techniques in a single system on the principle that there are different forms of "uncertainty".

The main competitors in the handling of uncertainty using numerical techniques are Bayesian (subjective) probability theory, Dempster-Shafer evidence theory and fuzzy logic and its cognates. Each requires an initial distribution of belief to be provided and offers rules of combination for calculating what the subsequent distribution of belief should be. As one would expect, there is no agreed method for the acquisition of prior probabilities (or fuzzy logic graphs).

Bayesian probability theory requires belief measures for the occurence of a piece of evidence, the truth of a hypothesis and the occurrence of a piece of evidence given the truth of a hypothesis. The combination rule - Bayes theorem - provides the belief measure of the truth of a hypothesis given that of the existing pieces of evidence.

Dempster-Shafer techniques require the distribution of belief measures across a frame of discernment consisting of hypotheses. However, belief measure can be attached to sets of hypotheses without commitment to the belief measure of any hypothesis in the set, as one might assert P v Q without asserting P or asserting Q. For this reason it is said that the theory permits an explicit representation of ignorance. By contrast the Bayesian approach would be commited to splitting the belief measure equally among each member of the set. The combination rule involves the intersection of these sets of hypotheses in a manner which requires the evidence to be independent. The output of the calculations is a support measure and a plausibility measure for a set; the difference between them is the evidential interval. By contrast the Bayesian techniques would provide a single figure.

Fuzzy logic requires one to provide graphs giving the degree of membership of a value in a set representing the extension of a fuzzy predicate. Its combination rules are based on taking maxima and minima of fuzzy set membership functions. Its output is (ideally) a graph representing the values of a fuzzy predicate which is the end result of some inference process.

Examples of the superiority of either Bayesian or Dempster-Schafer methods for particular areas keep appearing. Neither seems to have suffered any particularly damaging theoretical blow, though Nilsson has commented (1989) of Dempster-Schafer evidence theory and fuzzy logic: "we regard these techniques as temporary idiosyncracies". Data fusion applications show a preference for Dempster-Schafer theory, perhaps because the use of distributed sources of information (sensors) makes it easier to satisfy the requirement for independence of evidence - its chief drawback. Harris (1988) provides a strong case for Dempster-Shafer techniques in command and control applications: equating ignorance as regards an hypothesis with indifference to it (as required by the "maximum entropy" assumptions in Bayesian techniques) does seem inappropriate in a combat zone. However, he implies that the Bayesian approach is more consonant with the architecture of machine learning systems such as neural nets.

The simple evidence-hypothesis arrangement can be expanded to incorporate intermediate states thereby creating multi-level evidence-hypotheses relations in a belief network. Propagation of belief measures in such networks has been performed using both Dempster-Schafer techniques (Gordon and Shortliffe 1985), and Bayesian methods (Pearl 1986). This process can lend itself to parallel algorithms under certain circumstances.

Provan (1989) has adapated the label updating techniques of de Kleer's ATMS to aid the computation of Dempster-Shafer belief functions. His inital interest was to use probability information as a means of providing a control regime for the ATMS. Though there had previously been no apparent adoption of the idea of combining an reason maintenance system with probability handling, some belief networks with weighted arcs and update methods were presumably doing this in all but name.

Fuzzy logic continues to be used. The interest is pragmatic: it can make the knowledge acquisition process less painful and help present results more meaningfully. There is no lively debate on the role of fuzzy logic in solving philosophical puzzlers such as the sorites paradox or providing a system of vague inference. In the past, analyses of fuzzy terms and statements and proposed solutions have been more worrying than the problems, with naive analyses of linguistic phenomena being followed by complex formalism and techniques. Mathematics research on fuzzy sets, which are the heart of the semantics of fuzzy logic, is likely to be too abstract to have an effect on the knowledge representation issues.

The main obstacle to the use of fuzzy reasoning is computational feasibility: the storage and manipulation of the distributions can be prohibitive. Parametric representations of fuzzy graphs have been proposed. For example, with a simplified shape one can represent the interval for which the membership function has value 1, together with the left and right widths of the distribution. One demonstrator system using fuzzy reasoning (Roberts & Hughes 1989) introduces simplifying assumptions and requirements to obviate the need to store multi-dimensional matrices or to compute aggregations of the consequent; many fuzzy inferences describing position and speed could be made in real time, it is claimed.

The leaders in the fuzzy logic field are Dubois and Prade (1988). They have made developments to possibility theory, a cognate of fuzzy logic. It allows the representation of the possibility of having a value - "an elastic constraint on the value that might be assigned to a variable", as Zadeh describes it (1981). Roughly speaking, possibility graphs show values a variable might assume and to what degree any value, as opposed to the others shown, is likely to be the value of a variable; fuzzy set graphs show values for a continuous variable and to what degree each one belongs to the set.

Dubois and Prade suggest the use of possibility theory with default logic through using weights on defaults to rank them in order of typicality. This would provide a mark of uncertainty in the default or an estimate of the absence of exceptions, and allow a distinction between degrees of uncertainty and reasons for uncertainty. They claim their proposal is easy to implement in an object-oriented approach. No application has appeared incorporating this particular idea.

A recent system (Le Naour & Desjouis 1989) is described as using possibility theory for dealing with imprecise and uncertain battlefield surveillance information. Imprecision in a measurement is dealt with by mapping a range of values for a quantity onto a possibility distribution. Uncertainty in an <attribute = value> statement is represented by two indicators - a possibility and a necessity. (There are similarities to the support and plausibility couple of Dempster-Shafer evidence theory). The necessity seems to depend on the possibilities of existing alternatives rather than on the possibility of the negation of a statement itself.

There are no attempts to link fuzzy logic with the work on qualitative reasoning. In one way this is not surprising given that the basic strategy of fuzzy logic requires the mapping of qualitative terms to scale readings and fuzzy membership values. However, the interest in order of magnitude reasoning in qualitative reasoning work which attempts to refine a partial ordering ("smaller than" to "much smaller than", for example), overlaps with some of the early work done in fuzzy reasoning on linguistic approximation and comparative terms. There is the same interest in generating a vocabulary of descriptive terms and adverbs. A technique of "fuzzy instantiation" of frames has been used to drive a best-first search down a tree of descriptions to output a status report in more meaningful terms.

It is worth noting a certain use of more direct or analogical methods: using a space to represent ones uncertainty about spatial position. One positional fusion method uses shape intersection to compute an likely position of a vehicle, rather than feeding belief measures into some probability calculus. Another system for path-planning tests for the overlap of a line representing a vehicle path with an ellipse representing the uncertain position of a moving obstacle.

As regards the non-numerical approaches to uncertainty, few people claim to be using the method of Endorsements advocated by Cohen (1985) as a preferred approach and it is not an active research area. A system of Tobat, Rogers and Cross for C3 (1986) is described by its authors as using endorsement theory. This seems to mean that a discrete set of qualitative descriptions is used for the belief measures of evidence. Evidence is ranked using weighted matching against prototypes and applying a decision threshold to a measure of "distance to prototype". The richness of Cohen's suggestions, where rule-bases included a description of the quality and reliability of their constituents that could be transmitted with any solution they produced, is absent.

One major claim for Endorsement theory was that it did not compress the different supports which a hypothesis has into a single numeral lacking any meaning to the user of a system. It is possible that such a feature has been incoporated into AI programs through use of the various book-keeping operations of reason maintenance systems. Another non-numeric treatment of uncertainty identified by Cohen is the blackboard (control) approach, consisting of database partioning, evidence and hypothesis posting, and knowledge source activation by priority. No-one has been writing systems of this form from scratch with the prime aim of handling uncertainty; it usually arises from the use of toolkits facilitating such an architecture

The explanation based reasoning approach to uncertainty championed by Hirst (1989) has some similarities to the earlier notion of abductive reasoning. A good effect of the approach is to establish an ordering of (or a preference for) hypotheses based on the kinds of methodological criteria used in assessing scientific theories; it treats uncertainty as largely a consequence of competing hypotheses. However, numerical measures are re-introduced in the form of penalty values and the issue of run-time updating of the explanatory structures and penalty values is not tackled.

Another suggestion for capturing everyday reasoning about possibilities is Halpern and Rabin's LL (1987) - the logic of likelihood. It introduces a modal operator L (Lp = "p is reasonably likely to be a consistent hypothesis") and a semantics based upon the notion of state as a set of hypotheses. A limit operator L* can be applied to iterated modalities, and inference chains can be "diluted" to prevent the problems related to transitivity in conditional reasoning (the sorites paradox, for example) from arising. This may solve some toy problems but needs developing before it is a serious contender.

No applications work is using techniques such as Bundy's incidence calculus (1985), or Nilsson's probabilistic logic (1986), though they are frequently referenced in research papers. These approaches relate belief measures to the notion of sampling used in objective (frequency-based) probability theory. Again, the use of the notion of a "possible world" is prominent. Different valuations (interpretations, situations) which can be assigned to a theory constitute a sample space. The differing truth-value which a proposition has in these valuations is used to explain the meaning of the probability attached to it. The main problem with such "truth-value sampling" approaches to uncertainty is tractability.

No-one has taken up topics such as pleonetic logic (the logic of majority) or the treatment of generic sentences. Their connection with the handling of uncertainty is perhaps indirect, though the work has some relevance to arguments about defaults.

Of interest is the introduction of ideas related to distributed AI and distributed processing into the research articles on uncertainty and probability (and vice versa). This makes sense given that many of the reasons for wanting to introduce the handling of uncertainty into systems (especially) in the command and control field relate to the distribution of the information sources. A prominent researcher in distributed belief networks with parallel constraint propagation is Pearl. A recent model of Ng and Abramson (1989) for combining multiple knowledge bases drew on opinion pooling techniques from decision theory literature. In their model a central controller uses a confidence matrix to asses the contributions of the opinions of experts who had given themselves a prior expertise rating. The model is likely to be expensive to realise because of the knowledge acquisition effort required to obtain probabilities and the run-time overheads of the architecture.

No system has yet managed to combine the distinction between different sources of "uncertainty" (genuinely random systems, resolution of scales, measurement error and imprecision, vagueness of qualitative terms, existence of conflicting opinions, integration of information from disparate viewpoints, possibility of alternatives, under-determination of a theory by evidence, natural modesty, application of tentative models/theories, use of principles induced from experience) with a selection of different techniques for coping with them appropriately. There is an emphasis in most of the work on the use of probabilities to select a candidate from of set of pre-determined hypotheses. The temporal nature of the use of evidence and hypotheses has not been incorporated into any of the uncertainty calculi; this contrasts with the enthusiasm with which logical calculi are being adapted to the fields of non-monotonic and temporal reasoning. While updates to the degree of belief in a hypothesis are performed when evidence for it, or for its rivals, occurs, there is no work on how mere lack of any evidence at all for an interval of time should affect the status of an hypothesis, nor on how to weigh evidence which occurs during a narrower period of time than would be expected. These notions can be incorporated into the content of the actual hypotheses but at the price of losing simplicity. Issues such as the staleness and downgrading of hypotheses need to be addressed.

There is a lack of interest in the formation of opinion, both individual and collective. The field might benefit from studying individual human judgement, how people make up their mind in juries, debates and elections; how a scientific theory is applied and assessed. Presenting the inferences of a system in these terms probably has benefits at least equal to those coming from employing fuzzy logic to incorporate vagueness and qualitative descriptions. While studies of similar metaphors has been a fruitful area for research in distributed AI, the handling of uncertainty has been marred by a lack of richness in its concepts of judgment and belief.

3.1.6 Spatial Reasoning

Spatial reasoning is relevant to AI not as a quality present in knowledge (as uncertainty handling) nor as a problem of the use of knowledge (as reason maintenance systems), but as an element in the subject matter of many fields of interest. Charniak & McDermott use a threefold division in their brief treatment (1985) of spatial reasoning: route finding, "naive physics" problems, and design. Within these areas one finds the topics of relative object size, position, orientation, and composition; the direction, speed and planning of movement and the gleaning of 3D information from 2D representations. There are a number of AI related research areas which can contribute: logic, linguistics, cognitive science, qualitative reasoning, (intelligent) CAD, robotics,and vision. The underlying desire of much work is to be able to perform translations from graphical to textual information, or a processing of visual input to form spatial models, diagrams and sketches, followed by a transition to symbolic descriptions.

Spatial Logics ?

It is natural to approach an investigation of spatial reasoning in AI in the light of the extensive work done on temporal reasoning. There exist no spatial logics analogous to the temporal logics which one finds in research work and applications. No-one has seriously claimed that there should be. The reason is that time and tense play a more vital role in language than space and place. The former pair have a much better claim to be treated as "logical" elements (independent of subject matter) when the validity of arguments is assessed. It is possible, by analogy with the treatment of time, to interpret certain modal operators as "everywhere" and "somewhere", instead of "necessarily" and "possibly", (or "always" and "sometimes") and to extend an intensional logic to include possible worlds for spaces. Indeed, the spatial and temporal prepositions of natural language exhibit great similarities. However, there is a lack of problematic cases which suggest the need to develop the kind of logical systems found in the area of temporal reasoning.

The absence of a logic-oriented approach does not prevent some borrowings from the treatment of time from occuring in spatial reasoning issues. For example, Fleck (1987) notes the relevance of Allen's approach to the topic of events overlapping in time, to the issues of representing the boundaries between objects in space. Since much temporal thinking uses a spatial analogy - time as a line - one could consider these moves as the repayment of old debts.

There have been attempts to provide a "computational semantics of natural language expressions which describe spatial expressions" in the CITYTOUR system (Andre et. al.1986). These "semantics" have little relation to the models of temporal logics. Essentially, it involves developing an interpreter for a language (a query-answering system, for example) which relates prepositions such as "alongside", "past", and so on, to calculations involving object position, distance and motion. A similar facility is required in task-level robot programming. The interest of the natural language work consists in showing how the types and properties of objects involved in a situation can affect the evaluation of a spatial description. One sees that spatial knowledge is not so domain independent as temporal knowledge.

Axiomatics

There is one issue in cognitive science, the investigation or refinement of existing psychological theories with the aid of computer experiments, which has a bearing on the best approach to take in tackling spatial reasoning. This is the debate concerning the relative importance of "abstract propositional representations" versus "mental models". A more "propositional" approach will concentrate on the provision of general statements to represent knowledge about space, whilst a "mental model" approach would emphasise the importance of capturing the use of diagrams and reasoning by analogy.

In temporal logics, knowledge about time is largely resident in the semantics provided for the extended logical vocabulary. An alternative would be to state a theory (intended to be) about temporal relations in a set of axioms using the non-logical vocabulary of a standard logic. No special axiom systems to codify aspects of commonsense knowledge of space are exciting current interest. The fact that, for example, nothing can be in two places at the same time is left implicit in the algebra of a co-ordinate system, or unstated because of (implicit) use of closed world reasoning or unique naming assumptions.

Study of axioms systems and decision procedures for various mathematical spaces using algebraic methods is a thriving field. The seminal result seems to be Wu's method (1978) (cf. Kapur & Mundy (1988)) which simplified the task of automated theorem proving in algebraic geometry, though it is more restricted than earlier, more complex methods. It has been applied to the topic of perspective viewing in image understanding. One of the earlier methods (Collins 1975) has been used to develop algorithms for path planning.

Shape Grammars

There is one old technique for describing a certain kind of spatial knowledge which mimics procedures of logic and linguistics, but is not totally "propositional" in approach. This is the shape grammar of Stiny (1980) which has been used to axiomatise the "style" of a designer. Derivation or rewrite rules are given for shapes rather than sentences. It enjoyed a vogue in design circles some years back and a similar idea is now appearing in the AI literature as a process grammar for shapes, though apparently without knowledge of the earlier work. This process grammar has some bearing on the area of qualitative physics since, unlike the design work, it is more interested in understanding the process of transformation - what forces are at work producing the changes of shape - than the end result.

Cognitive Maps

Another form of spatial reasoning work involves the knowledge which one has of surrounding space. This is the field of "cognitive maps". It studies the manner in which people form the knowledge of the space of their environment ("large scale space") from necessarily partial views. The work seems primarily a cognitive science issue but relevant observations have emerged for robotics. Kuipers did some early work (1983) on what is involved in knowing a route. Using the two notions of a "view" and an "action", which is associated with the view and leads onto another view, he attempted to link the phenomenon of not being able to describe a route

without following it, to the complexity involved in the different patterns of association between views and actions.

Another prominent researcher in providing a computational approach for cognitive maps is Yeap (1988). He sees a useful distinction between a "global view" and a "global map" of space. The former concentrates on the connectivity of objects, having a sequential character useful for route finding; the latter is concerned with the layout of elements and is a kind of simultaneous view of the whole environment. A cognitive map sufficient for route finding need not bear too much relation to an accurate map of the environment, which would take more resources to compute. Yeap draws on Marrs work on vision (Marr and Nishihara 1977). One module transforms surface information into a 2-1/2 D sketch used to build a "raw cognitive map" which a second module uses to build a set of "place representations".

Qualitative Reasoning

The treatment of space in the qualitative reasoning field, as surveyed by Blackwell (1988) is limited. Qualitative systems for previously quantitative topics exist. There is a "commonsense arithmetic" and a "qualitative trigonometry", for example. The key features of such systems are the setting up of partial orders between objects. A qualitative term usually "undetermines" the numerical value of some object's quantity, leaving only rough bounds on the actual value. Consequently, attempts to make inferences with these qualitative systems results in a form of constraint based reasoning, with the constraints being used to set up the partial orders.

The order relation can be simple, such as would be used in proper quantitative systems ("less than") or more complex, involving order of magnitude descriptions ("much less than"). However, there are a number of representation and inferential problems associated with these techniques. One is that keeping an ordering of objects can require expensive computation when a new object needs to be fitted into the network. The problem is compounded the more complex the qualitative information is. It is comparable to the updating work which needs to be done by certain reason maintenance systems when new information comes to light. Another problem is the generation of meaningful vocabulary beyond an initial set of discrete qualitative descriptions.

As regards spatial reasoning in qualitative physics work, there is a bias towards the description of mechanisms and processes, but little on the description of motion itself, for example. Space is reduced to networks of "places" or a discrete set of regions. Any special knowledge which is employed in everyday reasoning is ignored because motion is not distinguished from any other state transition. It may be better to leave the structure of free space implicit in the operations and results of operations which can be applied to objects, rather than employing unsatisfactory methods of dividing it up and disguising its continuous nature. In sum, nodes and arcs are inadequate representation primitives for rich spatial reasoning. Blackwell developed two qualitative space representations: the Axially Specified Subparts and Features for a sliding problem, and the Extended Polygon Boundary for a path planning problem. More rarefied representational primitives and languages have been suggested. Goguen (1988), for example, presented an abstract and executable specification language for geometric constructions but its practical value remains to be seen.

Route Finding

By contrast to qualitative physics, problems where the main emphasis is on path planning do use a better representational strategy. The standard technique of partitioning space into smaller sub-spaces can use not just a grid but various polygonal cells. These are constructed, for example, from the line of sight of a "robot" at various positions, in relation to a number of obstacles in the space. Another approach is to construct a "configuration space" from the degrees of freedom of the robot with objects (obstacles) represented as algebraic surfaces in this space. The use of discrete regions is less damaging because of the narrower scope of the problem. Most systems in this area are concerned with finding a route whilst avoiding any stationary obstacles, as opposed to working among moving obstacles. Another division concerns the time when a route is calculated: is it worked out before any movement is undertaken or is it calculated incrementally?. Gilmore's system (1989) for terrain navigation by an aircraft generates "maps" out of "waypoints" and "route segments". A digital altitude map is used to plan an optimal vehicle route through a network using a best first search technique. (The nodes-and-arcs analysis seems appropriate in these circumstances). Tychonievich et. al. (1989) have investigated 2D incremental path-planing with moving obstacles, adapting the manoeuvring board method commonly used for nautical navigation. The accuracy of such incremental methods is dependent on the frequency of information sampling it needs. They also allow uncertainty in the position and velocity of obstacles through a spatial representation of uncertainty, rather than any probability calculus. A region, instead of a point, is used to stand for the area where an entity might be. This idea occurs in much constraint-based reasoning about space and the properties of spatial objects, for example, in Donald's work (1988) on error detection and recovery for a robot.

CAD

A richer set of primitives can be adopted from the CAD stream using the techniques of constructive solid geometry, boundary representation, wireframe modelling and so on. The basic entities of these techniques are themselves rich in spatial properties. Ironically, work in "intelligent CAD" (CAD with AI techniques) seems to have placed too much emphasis on AI representation methods. There is an interest in formalisms for the declarative specification and description of shape, with logic based description of object relation and composition linked up with drawings and diagrams. Work in this genre, for example at EdCAAD (Edinburgh University Computer-Aided Architectural Design unit), does not seem to have advanced very far. By contrast, the thinking at the Edinburgh AI department, whose EDS (Edinburgh Designer System) contains a module for spatial reasoning, favoured a hybrid approach: CAD-style geometric modelling combined with AI-style representations for more general object descriptions.

Analogical Reasoning

One motive for arguing for a richer set of primitive representations or a more "spatial" treatment of space is to allow reasoning by analogy - even if following up the spatial analogies is ultimately performed using algebraic methods. This can be seen as wanting spatial knowledge to be implicit in the right way. Gardin and Meltzer (1989) have complained that qualitative physics work is still too tied to the symbolic methods, albeit of a non-numeric kind. They proposed an analogical representation

of naive physics which in their initial work involves a pixel set representation of objects and substances. This seems related to the strategy which Forbus (1983) has previously criticised as a "naive analogue" position, so called because it leans on an analogy between the digitising elements of the human visual apparatus and the discrete elements of a computer screen. A basic criticism of the approach is that it can do justice neither to the low-level structures nor the high-level knowledge which humans possess. Forbus doubted the role of geometry theorem proving or algebraic manipulations in human knowledge of space and preferred to concentrate on the appeal of a diagram to a human problem solver. The topic leads back to the issues of "propositions" versus "mental models". A diagram has a number of advantages. It can be used to establish the consistency of a set of sentences and there is usually a natural mapping between it and the characteristics of a real-world problem. However, since it functions as a model for a set of sentences it can reduce the indeterminacy in the description of a problem. This is a mixed blessing. Some indeterminacy is valuable if it keeps alternative interpretations under consideration, or stops one getting involved in the exact details of a particular interpretation and the calculations which it might involve. This is claimed to be one benefit of a qualitative approach to quantitative matters. On the other hand, the intellectual effort of coping with all possible interpretations or ensuring that a reasoning process remains strictly abstract can be too great to allow any conclusions to be drawn in a reasonable amount of time. Willoughby has investigated the power of the intuitive principles which a diagram or drawing contains in enhancing qualitative reasoning; his Q-graph builds on Kuiper's QSIM (1986), a tool designed to investigate qualitative physics problems.

Summary

The essential weakness of AI work in spatial reasoning is that it is done relative to an application - this may be unavoidable. There are no ongoing debates in the literature comparable to those on temporal reasoning, because of a lack of problematic cases. Work on automated theorem proving for geometric systems is dominated by algebraic methods and still needs to be incorporated into more everyday reasoning tasks.

3.1.7 Planning

Many attempts have been made to design planning systems in various domains using AI techniques. As usual these have taken the form of domain specific systems (e.g. in Air Traffic Control and mission planning) encapsulating expertise from human planners, and domain independent systems which have tried to produce general principles for planning systems. A paper by Austin Tate, of the Department of AI at Edinburgh University, provides a summary of the state of the art in the latter category of system (Tate 1985). Some of the most significant developments in the field are highlighted below:

Non-Linear Planning

The linearity assumption in planning is that goals are essentially independent and can be solved in sequence without too much concern for their interactions. This assumption sufficed for the very early systems but it was soon found that practical problems, even in the classic block-stacking domain, put linear planners into trouble as early commitments to solve one goal prevented later goals from being satisfied. In non-linear planners a "least-commitment" approach is adopted so that actions will take place, and goals will be satisfied, in parallel, until some decision has to be made about sequencing because of common demands on resources or to exploit commonality in

requirements between tasks. Commitment problems do still arise in non-linear planners because choices are still being made about actions, and those choices may need to be backtracked over.

Critiquing

The idea behind critiquing is that while the main body of a planner is dealing with the search for appropriate actions, "critics" are looking at the evolving plan and suggesting localised improvements to it. Critics can be used for many purposes, including detecting the goal interactions referred to above or optimising resource usage, etc.. Critics can either directly alter the plan structure themselves or propose things for the planner to consider. The idea is a nice one from the programming perspective as it provides a clean mechanism for encapsulating "housekeeping" procedures, and from the knowledge engineering perspective, since it is often easy for people to say what is wrong with a plan rather than how it should be generated in the first place.

Hierarchic Planning

Most planning systems operate hierarchically; they break down the solution to one goal into a number of simpler subgoals and then try to solve each of them, which may lead to further decompositions. Planners do differ in how they expand or solve goals at different levels, some doing it strictly by level and others more opportunistically. Typically it is better to get a working "rough cut" at a plan, i.e. at a high level, before elaborating lower levels, since backtracking a decision at a high level will involve, usually, discarding a lot of the lower level work. The hierarchic decomposition of the planning problem maps readily onto how most expert planners would view their method of solution.

Meta-Planning

Unlike simple hierarchic planning, a Meta-Planning system (Stefik 1981) reasons about the planning problem at several levels. It separates out and makes explicit the control knowledge that directs the planning process, and creates a layer in which decisions about that process can be reasoned about. At the domain level the planning can still be carried out hierarchically, the meta-level removes the implicit and possibly sub-optimal search mechanism that performs the planning.

Skeletal Plans

In many applications the responses required from a planner can be seen to be very stereotyped. For example, emergency reactions are particularly well constrained, there being, usually, a lengthy set of procedures to be followed. To capture this phenomenon some planners have used a set of "skeletal plans" which predefine the type of response and much of its form. The planning process then reduces to the selection of an appropriate skeletal plan and the tuning of it for the specific circumstances in which it will be used.

Action Specifications

In order to reason about satisfying goals, the planner must be able to understand what the effects of any action are going to be. Several "languages" have evolved for doing this, but each is based on the idea of specifying a set of conditions that will hold after an action, given a set that holds before it. Actions are generally specifiable at several levels; at the lowest, the primitives are directly executable by the planner; at higher levels more sophisticated actions are specified that may themselves expand to many primitives. Specifications typically have variables that can be bound, so that, for a robot, the action *pick_up(X)* would have a pre-condition *empty(hand)* and post-condition *in_hand(X)*. Extensions have been made to such basic facilities to allow partial ordering of sub-actions, constraints on variable bindings, action durations and uncertainties to be expressed.

Opportunistic Planning

In contrast to the hierarchic planning model of successive refinement of action to form a plan, the model developed by Hayes-Roth (1979) assumes people's planning activity is largely opportunistic. That is, "at each point in the process, the planner's current decisions and observations suggest various opportunities for plan development". Psychologically there is quite a lot to justify this model, and it bears more than a passing similarity to Suchman's idea of Situated Action (Suchman 1987).

There have been other types of representation recurring through the design of various planning systems, particularly to simplify the checking of goal conditions and monitoring interactions between parts of plans.

Real-Time Planning

We discussed above the development of what might be termed "conventional planners" from the AI standpoint, i.e. those dealing with closed worlds, perfect execution of actions, perfect sensing, and no time constraints. These have formed the mainstream of AI research for fifteen years, with relatively little attention having been paid to real-world planning problems in Command and Control or elsewhere. Over the last few years, however, there has been considerable interest in the development of reactive planning systems, and the debate over the best approach to their development has reached the height of a collection of point and counter-point articles in the AI Magazine (1989).

The contested notion is whether or not a planning system can pre-compute or be given all the plans it will need to react to all the circumstances it may find itself in. The debate is somewhat sterile because what people had actually been suggesting was the caching of "Approximate Universal Plans" which are instantiated and elaborated in a situated manner. The rationality of this had been anticipated in the Command and Control community many years earlier, when a system by Markosian and Rockmore (1984) used skeletal plans to allocate resources. This use has been repeated many times since in UK-developed systems such as the TARA (Threat Assessment and Resource Allocation) system at RSRE (Bennett 1985) and, more recently, Plessey's Command Support Tool (Lingard et al 1988). Somewhat unfortunately these planners have been referred to as 'Script' based rather than skeletal plan based. The Script was a schema used by Schank in situation analysis rather than a planning mechanism per se. In the aforementioned systems, the planner has a skeletal plan

store and a mechanism for selecting one depending on the situation; the selected skeleton is instantiated and fleshed out in a manner appropriate to the specific circumstances before being acted upon.

In simple systems, particularly the average one or two scenario demonstrator, the skeletal plan approach works well. The planner's skeletons happen to correspond well with the scenarios. The problem clearly lies in defining sufficient skeletal plans, being able to recognise the circumstances in which they are applicable and, more importantly, the conditions when none of them are actually appropriate and the need therefore to generate plans from scratch.

There are other approaches to the time-constrained planning problem, Georgeff and Lansky (1987) discuss PRS, their Procedural Reasoning System, which interleaves planning and execution and appears to be an intermediate architecture between planning systems such as STRIPS, NOAH and NONLIN and the skeletal type; Georgeff and Ingrand (1989) discuss guaranteed reactivity within PRS. Dean and Boddy (1988) propose an architecture based on 'anytime algorithms', Kaelbling (1986) similarly suggests an incremental approach to planning based on a declarative language called REX. Clearly approaches can be integrated such that a skeletal plan can be used as a default and, as time permits, a situation specific plan is built.

Though the skeletal plan approach provides a neat mechanism for reactive planning it provides no inherent mechanism for handling the uncertainty associated with plan execution. There are many facets to this issue, including the monitoring of plan execution, detection of plan failure, debugging of plans and plan repair. It embraces too the ideas of predictive and contingency planning, i.e. anticipating potential failures and consequently making plans robust by caching plan repairs.

These are difficult problems which have waited for the practical deployment of planning and scheduling systems before they have been addressed. Ow et al (1988) examine constraint conflicts within the OPIS scheduling system and experiments with schedule repair methods. The IPEM, for Integrated Planning Execution and Monitoring, system at Essex (Ambros-Ingerson and Steel 1988) is a framework for those functions using a simple production rule formalism to control the planning process. Adey (1987) describes a NONLIN-based planning system for a naval domain and addresses Conditional, Contingent and Deferred planning issues (and the problems of implementing them in NONLIN). Wood (1988) describes an architecture addressing planning in a dynamic domain but furnishes few details of representations and mechanisms.

While this gives a cursory overview of real-time planning, such factors as counterplanning and distributed planning, which are inherent within command and control problems, are also the subject of much research. For example see Carbonell (1981) on counterplanning and proceedings from any of the recent workshops of the Planning Special Interest Group for evidence of the growing popularity of research in distributed planning.

It is fairly safe to conclude that planning in a command and control context is a very long way from being a solved problem from the AI standpoint. It is a relatively simple task to build a demonstrator around skeletal planning techniques but this has severe shortcomings as a basis for a realistic ship-board demonstrator without an investigation into the more fundamental issues involved in either naval or TMD operations planning.

3.2 Activities and Resources

3.2.1 Temporal Reasoning

Activities

The most widely published author on temporal reasoning within the UK is Tsang of Essex University. His work can be considered of world status in the area. Other work is being done at Edinburgh and Nottingham on modal logics for temporal representations. Imperial College is involved in Esprit projects on temporal reasoning.

There has been a small amount of work by software companies in applying explicit temporal representations. Ferranti are involved in the EQUATOR ESPRIT project on qualitative temporal reasoning; CCL have incorporated a temporal representation scheme into MUSE.

Availability of Resource

Again there is some concern over access to workers at Essex, but otherwise resources are as above.

3.2.2 Modal Reasoning

Activities

There appears to have been little take up by software companies of modal logic work. The most prominent research within the UK would appear to be at the Universities of Essex, Edinburgh, Nottingham, Leeds and Imperial College. Hewlett Packard have the most active industry-based group working in the area.

There is a great deal of activity in the U.S. Acknowledged centres of excellence are the University of Rochester, Xerox Parc - (contact J de Kleer, D Bobrow, P Hayes), MIT, SRI, CLSI

European activity is developing rapidly in this area (including UK participation), through the ESPRIT and BASIC research programmes. Important work includes

- ESPRIT Project No 973 ALPES - Advanced Logical Programming Environment
- ESPRIT Project No 527 Communication failure in dialog
- BASIC Porject No 3178 REFLECT - Reflective expertise in KBS
- BASIC Project No 3126 MEDLAR - Mechanising deduction in the logics of practical reasoning

Availability of Resource

As above, little evidence of specific expertise.

3.2.3 Model-Based or Deep Reasoning

Activities

UK Activity

The majority of work in this area is University based. The following sites are known to possess useful resources in model based reasoning.

- University of Sussex (Cognitive Studies contact Prof A Sloman)
- University of Edinburgh (AIAI contact Prof A Tate)
- University College of Wales (Comp Sci contact Prof M Lee)
- University of Aberdeen (Comp Sci contact Prof J Hunter)
- Herriot-Watt University (contact Prof R Leitch)

Other Activity

There is a great deal of activity in the US. Acknowledged centres of excellence are:

- University of Illinois (Comp Sci contact Prof K Forbus)
- University of Texas (Comp Scu contact Prof B Kuipers)
- Xerox Parc (contact J de Dleer, D Bobrow, P Hayes)
- MIT
- SRI

European activity is developing rapidly in this area (including UK participation), primarily through the ESPRIT programme. Important work includes:

- University of Amsterdam (Comp Sci contact Prof ten Hagen)
- ESPRIT Project No 2409 EQUATOR Environment for qualitative temporal reasoning
- ESPRIT Project No 820 QUIC Qualitative Industrial Control

Availability of Resource

With regard to command and control applications, AIAI probably have the most familiarity with the issues and would therefore be a logical source. University College of Wales and Herriot -Watt would be further choices.

3.2.4 Reason Maintenance Systems

Activities

Again many universities have studied Reason Maintenance Systems, often as part of a larger project activity. Edinburgh, for example, worked on such a system for the Alvey DTP demonstrator, Leeds were involved with Software Sciences on the work for ARE on FLYPAST. In a similar fashion, RMS or ATMS systems have been built by software houses for their systems. Two such mechanisms are available for MUSE.

Availability of Resource

Software Sciences and CCL have direct experience, to our knowledge, of building such systems. In the latter case this applies to a real-time object oriented environment too.

3.2.5 Uncertainty

Activities

The most prominent workers in this field are probably those on fuzzy logic at Queen Mary College and Bristol University. There does not seem to be a prominent centre progressing the state of the art in this field, though uncertainty plays a part in much work in, for example, robotics.

Defence contractors are possibly greater sources of expertise in this area than academic institutions since they have been involved in building pattern recognition and classifier systems for military applications for many years. RSRE has long been a sponsor of work on uncertainty, and development of methods such as Dempster-Shafer for command and control related areas.

Availability of Resource

CCL, SD-Scicon, Logica, Ferranti, SSL are all known to have been active in this area.

3.2.6 Spatial Reasoning

Activities

Work in the Design, Vision and Robotics fields has been the main source of spatial representation work worldwide. Strathclyde, Oxford, Cranfield, Turing Institute and Edinburgh are therefore sources for this kind of work. There appears to have been little work on symbolic approaches to spatial reasoning.There has been some work work of a psychological nature on human spatial reasoning at Sussex University which is possibly of more relevance to the requirements in Command and Control.

Availability of Resource

There is little evidence of symbolic spatial representation work amongst the software suppliers. A recent CCL recruit obtained his MSc for research in Qualitative Spatial Reasoning.

3.2.7 Planning

Activities

AIAI has long been regarded as the UK centre of excellence in this field, and is a world class player. Most prominent academic AI departments have some planning work, with Essex, Strathclyde, Imperial and Sussex showing most strongly. Sussex has been doing work on real-time planning systems, but none have been tackling tasks of the complexity of command and control problems.

Most of the defence contractors and systems houses have done work in planning tasks associated with command and control problems.

Availability of Resource

As above

4 HUMAN COMPUTER INTERACTION

4.1 State of the Art

In the early stages of Sanderling the HCI state-of-the-art was directed at four sub-topics, viz, interface paradigms, user modelling, man-machine partitioning, and KBS Issues. With regard to the first of these, the state-of-the-art of controls and displays was represented by high-resolution, bit-mapped displays and the "mouse" pointing device which enables "direct manipulation" of information on the screen; the dialogue state-of-the-art was represented by the WIMP interface. On the subject of user modelling a number of modelling techniques such as GOMS, TAKD, and CCT were identified. With regard to man-machine partitioning, the state-of-the-art was represented by the human factors methodology embodied in Defence Standard 00-25. The state-of-the-art for KBS Issues; expert systems and adaptive interfaces was represented by manual knowledge elictation techniques and explanation facilities which re-stated the rule-base, and, on adaptive interfaces, little progress to report.

From Working Paper 3 onwards we have found it necessary to look at the choice of sub-topics afresh and have produced a different set, increased to six in number and reflecting a slightly different perspective on the subject. (ie physical interface, design methods and tools, modelling issues, cognitive issues, user support and organizational issues). Before considering the state-of-the-art of these sub-topics there is a general point on HCI to note. That is, given the objectives of the application areas (ie OpDep, EnFun and TMD), that there are two possible facets to HCI: the design of the HCI for a KBS, and the design of a knowledge-based HCI. We would maintain, however, that the two HCIs are in principle the same. What differs between KBS and conventional systems is the type and quality of information that can be made available at the interface. This point needs to be borne in mind if one is tempted to dismiss HCI research as lacking focus on knowledge-based systems.

As stated above, the physical interface state-of-the-art is represented by the ubiquitous CRT display and keyboard devices. The design of the physical interface is a subject that has generated a great deal of research and associated literature. There are now various human factors textbooks on the subject (e.g. Salvendy 1987) and standards such as Defence Standard 00-25. With regard to dialogue design there many guidelines available (e.g. Smith and Mosier, 1986; Williges and Williges 1984; Shneiderman 1987). The state-of-the-art is perhaps best represented by MOD/DTI guidelines on the subject (HUSAT, 1988) which were prepared under contract to A.R.E for application to the design of naval C&C systems. These guidelines strongly recommend, for example, that human-computer interfaces are provided with facilities to enable them to be operated in all three major types of dialogue (typified by menu selection, command language and form-filling respectively) as far as it is practicable and relevant to the task. "This is to prevent restrictive decisions by Designers as to which form of interface is best when, in practice, all are potentially useful" (4/A11, p.103).

On the subject of design methods and tools, the state-of-the-art is best exemplified by, again, the MOD/DTI guidelines (HUSAT 1988). Alternatively there is the work of Williges et al (1987). Their top-down design approach consists of three design stages including iterative steps throughout. Stage 1 is the Initial Design stage in which the software interface is specified; (entailing steps of design objectives, task/function analysis, focus on users, design guidelines, and structured walk-through). Stage 2 is the Formative Evaluation stage during which the software interface evolves in iterative fashion; it deals with techniques for obtaining user feedback to aid the designer in making decisions (eg rapid prototyping). Stage 3 is the Summative Evaluation stage and is used to test the final design configuration to ensure that it is functioning properly.

With regard to HCI design tools the state-of-the-art is represented by a number of visual programming tools. SL-GMS, `Trillium' (Henderson 1986) and `INTER/P' (Hashimoto 1987), for example, enable the prototyping of control panel user interfaces for devices like photocopiers and printers. They supply a limited range of interaction techniques, but are effective within that range. `Fabrik' (Ingalls et al 1988) has a richer set of components and can be used to design human-computer interfaces. On the subject of user interface management systems (UIMSs) a number of commercial systems exist (eg Apollo's Open Dialogue). Other tools are TIGERS and VAPS. The latter supports (so it is claimed) rapid, fully dynamic system evaluations and changes, reducing the need for explicit programming of the HCI. The user interacts, through a WIMP interface, with several VAPS modules to build a prototype (BAe (Military Aircraft) Ltd. at Warton use VAPS.).

With regard to modelling issues, the state-of-the-art is represented by a number of well-publicised methods such as Task Action Language (TAL), Command Language Grammar (CLG), Goals, Operators, Methods and Selection rules (GOMS), Task Action Grammar (TAG), Task Analysis for Knowledge Descriptions (TAKD) and Cognitive Complexity Theory (CCT). The differing modelling techniques reflect different aspects of HCI but none present a complete view. TAL addresses software, possibly hardware, and tasks; GOMS addresses users and tasks, as does CCT and TAG. In addition, the modelling techniques differ in their specific purposes. Some aim to specify the design of the interface at its initial stages of development (eg CLG) whereas others aim to evaluate an interface that already exists. TAL, for example, was used to compare the design of two versions of an interactive graphics system. Another purpose of some models is to predict user performance, the prime example being GOMS.

It should be noted that what most, if not all, these models have in common is that they use a formal grammar to represent the task. That is, the task is described using a special (symbolic) notation and associated rules, in the same way that linguists and computer scientists use grammars to describe languages. Moreover, because the models purport to describe not only the task per se, but the user's mental model of the task, the modelling techniques are often labelled as "user models". The psychological hypothesis behind CLG, for example, is that CLG describes the user's conceptual model of the system.

On the subject of adaptive interfaces, the state-of-the-art is represented by Alvey project "AID: Adaptive intelligent Dialogues" (MMI/HI/006). This was concerned with developing tools that would enable designers to build interfaces that adapt to the needs of an individual user or group of users. That is, the computer should be able to change the style or form of dialogue dependent on the varying needs of different users and the changing needs of individual users over time. To date, work in this area does not appear to have made great progress. It has been suggested that the types of information (ie the knowledge base) that the ideal adaptive or embedded user model might contain include:

- knowledge about a user's level of competence with a computer including a log of previous interaction.
- knowledge about the user's level of task expertise.
- knowledge about the usr's interests, values, aptitudes, goals, expectations, and assumptions.
- knowledge about the preferred method of interaction.
- knowledge about the user's mental model of how the computer works (ie his system model)

To summarise, the state-of-the-art concerning these models or modelling techniques is that it is probably premature to claim that they can in themselves produce optimum human-computer interfaces. (It is the case that interfaces that have been designed using available modelling techniques have not been particularly usable or efficient!). The modelling techniques do, however, focus the designer's activities onto the interface in a more principled and directed way than might otherwise occur.

The state-of-the-art with regard to cognitive issues must cover a very wide range of topics, viz, information processing; problem solving, reasoning, decision making; explanation; and mental workload. Indeed, because "cognitive issues" impinge on so many scientific disciplines, the 1980s saw the emergence of a new field--that of cognitive science. This is an umbrella term for once disparate approaches such as cognitive psychology, linguistics, computer science, AI, mathematics, and neuropsychology. The theoretical language of cognitive science is that of computation and information processing. The objectives of cognitive science are to define, build and test information processing models of the various sub-systems (and of their} sub-systems) making up an intelligent "agency", whether human or artificial, and eventually to make them fit together into general cognitive theories and systems (Bernsen 1989).

On basic models of human information processing the state-of-the-art is still probably represented by the the Model Human Processor (MHP) of Card, Moran and Newell (1983). The MHP is divided into three interacting sub-systems: the perceptual system, the motor system and the cognitive system, each with its own memories and processors. Associated with the memories and processors are various parameters and in addition a set of "principles of operation" (eg Fitts' Law). The model would appear to be unique in that it combines the numerous empirical results from cognitive psychology into a single model. Another current model is the multiple resources theory of Wickens (1984).

With regard to problem solving, reasoning and decision making, there are perhaps three lines of research which effectively represent the state-of-the-art. First, there is the work of Rasmussen and his co-workers at the Riso National Laboratory in Denmark (eg Rasmussen 1976, 1986a, 1986b). The focus of their research has been the analysis of operators' decision making, particularly as related to the design of control and safety systems for industrial process plants (Rasmussen's work is highly regarded (see for example Goodstein, Andersen and Olsen 1987) and he has played a pivotal role in establishing HCI aspects of ESPRIT and other research programmes.).

Second, there is the AI-related work on problem solving which has recently seen the development of SOAR (Laird, Newell and Rosenbloom 1987). In addition to the well-established ideas about problem solving being construed as a process of search in a problem space, and search being controlled by general problem solving methods, two aspects of SOAR are felt to be new to symbolic systems: universal subgoaling and learning by chunking. The former is a set of mechanisms by which the SOAR software detects an "impasse" - a point at which it can no longer proceed in its search using specific knowledge. At this point it must generate a problem solving goal through which it can fall back on more general problem solving methods. The learning by chunking mechanism is the complement of universal subgoaling, that is, when SOAR manages to solve a problem it compiles and stores the specific details of the solution it found as a new production rule. Then, should SOAR find itself in a similar situation this production rule can be evoked directly, avoiding the impasse and the need to solve the problem again. SOAR is claimed to offer "an architecture for a system that is capable of general intelligence".

The third area is that of reasoning and judgemental biases. This has been an area of study in psychology for decades. On the subject of reasoning Johnson-Laird at the MRC APU in Cambridge and Evans at South West Polytechnic have been very influential in recent years (eg Wason and Johnson-Laird 1972, Johnson-Laird 1983, Evans 1982). Much of the impetus for the study of users' cognitive strategies in judgemental tasks can be attributed to Tversky and Kahneman's demonstrations of certain judgemental heuristics. Evans has, however, extended the work in both these fields and is very much representive of the state-of-the-art (see Evans 1989a, 1989b).

On the subject of explanation, particularly the ability of expert systems to justify their expert advice, the state-of-the-art can be said to be re-statements of the inferential or production rules that led to the decision, or advice, with little consideration of "user-friendliness". Explanations are typically an audit trail of the rules that were used in coming to a conclusion. The situation is somewhat ironic, as pointed out by Cleal and Heaton (1988), when so many attempts to justify the value of expert/KB systems revolve around their ability to provide explanations!

Finally, in the area of cognitive issues, there is the topic of mental workload. This has generated considerable interest and empirical research for least the last twenty years. Most of this research has been directed at the development of techniques of workload assessment. The first point to make about the state-of-the-art is that a variety of useful techniques are available; AGARD (1987) presents a recent review of the field. Secondly, it is generally agreed that no single measure will be able to index workload in a variety of situations; the joint use of a number of assessment techniques will be required. Thirdly, and with regard to workload prediction techniques, it is evident that although they are potentially very useful (because they may be applied early in the system design process) they remain a rather neglected area of research.

4.2 Activities and Resources

4.2.1 Physical Interfaces

UK Activity

- Loughborough University of Technology CHI (LUTCHI) Research Centre (Alvey and other projects)
- Scottish HCI Centre (Alvey and other projects)
- HUSAT
- UK Companies (eg Logica Cambridge, British Telecom)

Other Activity

- ESPRIT 1-HUFIT project (385)

Availability of Resource

UK activity outlined above.

4.2.2 Design Methods and Tools

UK Activity

- University College London (Prof. Long)
- Queen Mary \& Westfield College, University of London (Johnson)
- HUSAT Research Centre
- Defence Companies (eg Plessey Research) and MOD establishments (eg A.R.E.)

Other Activity

- Medical Research Council APU (Young, Barnard)
- ESPRIT 1 and II projects
- C.E.G.B.

Availability of Resource

UK activity outlined above

4.2.3 Modelling Issues

UK Activity

- HUSAT Research Centre
- University of London (UCL \& QMW Colleges)
- UK Companies (eg Logica Cambridge, BAe SRC, Data Logic)

Other Activity

- ESPRIT I and II projects
- M.R.C. APU

Availability of Resource

UK activity outlined above.

4.2.4 Cognitive Issues

UK Activity

- M.R.C. APU (Baddeley, Broadbent)
- University/Polytechnic psychology Departments (eg South West - Evans)
- MOD establishments (RAF IAM)

Other Activity

Availability of Resource

UK activity as outlined above

4.2.5 User Support

UK Activity

- Computing Companies (eg ICL, IBM)
- Various Universities/Polytechnics
- National Physical Laboratory (NPL)
- Other Companies (eg Logica Cambridge, British Telecom)

Other Activity

- Various "usability laboratories" set up by computing companies (eg Olivetti's during ESPRIT HUFIT project)

Availability of Resource

UK activity as outlined above

4.2.6 Organizational Issues

UK Activity

- HUSAT Research Centre
- Various Universities/Polytechnics
- Tavistock Institute

Other Activity

- A.C.A.S. Work Research Unit

Availability of Resource

UK activity as outlined above

5 DATABASE / KNOWLEDGE BASE INTEGRATION

5.1 State of the Art

Coupling

Very few attempts have been made to integrate conventional DBMS with KBS within the context of real-time systems. Most approaches to real-time applications have concentrated on specialised software (eg. MUSE, G2). In these approaches, the database is kept in memory and the developer has to assume a high level of responsibility for the control of inference (Salle and Årsen 1989). An alternative approach involves writing one-off real-time storage and access mechanism. Using existing KBS and DBMS components it is not possible to implement 'active' data / knowledge bases where the input of a piece of sensor information automatically triggers appropriate inference.

Software Sciences Ltd. are of the opinion that a relatively straightforward connection could be made between DIOMEDES and a KBS tool such as NEXPERT. DIOMEDES is a real-time distributed architecture (at least partly implemented) that supports the use of 'triggers' by keeping monitors on specified input information. The system has the advantage of using an extended SQL as the query language.

If we assume that use of an existing DBMS is possible, then it is instructive to look at the variety of non real-time methods of coupling. KBS shell vendors refer to 'loosely coupled' systems meaning just the use of an intermediate file, via which data is transferred into the KBS internal representation, under the control of an external program call. 'Tight' integration means the ability to reference database entities directly from rule predicates, without the developer having to specify the access routes each time. Such systems can support the automatic generation of Data Manipulation Language.

The following components of current KBS shells are of importance in understanding performance :

RETE-like structures for run-time efficiency;

Optimisation with respect to specific application requirements;

Secondary memory management strategies;

Internal B-tree and record-based data management structures.

The above are not by any means well-understood areas, an example being the effectiveness of RETE-like structures (Forgy 1982) in the context of large rule sets and highly varying data. It is worth noting the current emphasis is on production rule-oriented systems, so there is even less understanding of areas such as the integration of Logic programming or Complex object manipulation.

Turning from existing KBS Shells, to one-off solutions, performance is affected by the same factors as are listed above, but there is an extra degree of freedom, since the inference engine can be optimised with respect to the application requirements. In addition it can be implemented without any of the redundancy that the KBS Shell approach requires. There are a relatively limited number of logical structures needed to support different functionalities (combinations of backward and forward-chaining, hypothetical reasoning, TMS, Blackboard Mechanisms), but a great variety of possible implementation details, especially with respect to binding and data-storage.

AI-Enhanced Database

Much academic interest has focussed in the clean integration possible between Logic Programming languages such as PROLOG and relational databases (Frisch and Allen 1989). Current demands upon database systems have resulted in the extension of the relational model to include complex objects. This is largely a market-driven process with DBMS vendors taking a central role. The motive is to encompass the diverse information requirements of office automation data, graphical objects and hierarchical data. It is unclear how the relaxation of First Normal Form restrictions (that the data items are atomic) will impact database architectures and performance, still less the architectures of distributed databases. The approach often has questionable theoretical underpinnings, and it is much more oriented towards the extension of existing systems (eg. POSTGRES).

In contrast, current research on Deductive Databases is aimed at a more theoretical understanding (Gallaire et al. 1989). The UK is well represented in this field. Areas of current research include the handling of transitive closures and recursive query processing. System performance targets are often very modest and there is wide acceptance amongst practitioners that Deductive Databases may not contribute to practical system building for 5-10 years. In order to use databases effectively as models, some research is taking place into temporal logics, which aim to provide effective ways of reasoning about patterns over time.

Intelligent Front-Ends / Natural Language

There are two main areas of research. One is driven by the need to support and manage databases that are distributed across more than one platform, and the other stems from the special needs that the addition of an inference component brings. The support of distributed databases is happening in a pragmatic and ad hoc way as a number of experimental and commercial (eg. Ingres) systems are being fielded. Work specific to KBS / Database integration is more complex.

Given a body of data, and an inferencing component for deriving more 'data' or 'knowledge', a user interface should decide how to most efficiently deal with a query. It may be that the query contains a discrete inference and query component, or that there are alternative routes to the same information. Either way the system should decide how to best decompose the query, and shield the user form the underlying implementation details.

Adding intelligence to the query language can reduce the amount of data being retrieved as well as providing more appropriate responses. This involves the use of application or domain knowledge to provide the higher level semantic validation. Such data-filtering techniques may prove essential if the user is not to be bombarded by a mass of complex spatial and temporal information. The communication of such information may only be possible using diagrammatic means, and this requirement may go some way informing the choice of storage mechanism.

User-modelling techniques have been used as one approach to accessing complex databases, improving overall performance of the 'system' (man and machine) by understanding the context of the query. Current database system employ techniques of varying sophistication to try and spot 'expensive' queries and alert the user to this fact. Such techniques might or might not prove to be easily extended to systems with an inference component.

Novel Database Structures

This sub-stream is concerned with the support of non-relational objects, but unlike the AI-Enhanced Database sub-stream is concerned with models that are not so much extensions to the relational concept, as fundamentally different forms that necessitate novel design of database. Object-Oriented databases are an obvious example (Garvey and Jackson 1989). A number of commercial OODB are coming on to the market (eg. Ontologic's ONTOS). The goal that at least one of these vendors states is of *increased* performance in certain applications. This increase in performance is based upon the use of a data representation that is more appropriate to the task in hand. An area for research is the extent to which the structuring of the data in object-oriented terms detracts from its use for a variety of applications having different requirements of the data.

Other aspects for study concern the possible negative aspects of having a richer representational structure. In relational databases data is commonly stored together that has similar patterns of retrieval, update, paging. The problem arises in OODB of the large object boundary problem. An object may display a lot of inter-connectedness. Once this object is used in a multi-user system, concurrency control may involve locking extremely large parts of the database (using pessimistic methods) or concurrent updates being aborted (using optimistic methods). An additional area of research is the extent to which more complex data models lend themselves to being distributed across different platforms. Existing distributed databases can make extensive use of the fact that a subset of a relation is still a relation. It appears essential that OODB provide the application builder with a high degree of memory control. It is currently unclear whether this presents large obstacles or not (and if so, for what types of application). Research into the feasibility of high transaction rate, distributed OODBs is therefore desirable. This activity lies at the implementation-end of spectrum, largely ascertaining the maturity of existing software products.

Dynamic Databases

The integration of dynamic databases with KBS presents a large number of challenges in the performance and contention area (Laffey et al. 1988). These surround the ability of the system to assimilate large amounts of frequently changing data and also retract facts, or terminate lines of reasoning in the face of more up to date or significant data. The database support for adequate concurrency controls (eg. optimistic locking strategies) is one area of concern. Most current real-time databases either make use of existing DBMS but with very large amounts of buffering, or use one-off solutions that put much of the responsibility for memory management within a program, and use the database for historical tracking. In view of the ad hoc of much of the work being done in this area, there are very few prescriptive models for how should proceed. General rule-of-thumb approaches include prototyping with the application of heuristics to guide the developer through a range of memory-management strategies. Whatever the proposed architectures for TMD and TDS, it is important that they are evaluated by non-KBS real-time specialists to gauge their practicality.

The implementation of KBS components accessing this data may itself place extra-ordinary demands upon the system. A 'variable precision' logic designed to yield results in a fixed amount of time may result in control moving unpredictably around in a problem space, causing a great amount of paging, and even further reducing performance. For any given application, that is a sub-set of the overall problem, it is often an open question as to how its performance characteristics will change as the application is scaled up. The theoretical limit will involve not only the bandwidth of the mechanism passing data between primary and secondary memory, but also the structure of the application itself. Work in this area can derive performance models that will assist in the assessment of technical feasibility of different software architectures. Simulation of different architectures is a useful method of establishing possible problem areas, in the absence of formal models.

Knowledge Base Management Systems

The term Knowledge-Based Management System has been coined to cover research into the convergence of database and knowledge base technology (Brodie et al. 1986, Myopoulos and Brodie 1989). It seeks to integrate the storage of data, facts, knowledge, production rules and logic languages in an overall framework that support secure, concurrent, multi-user access. Work in this area provides the 'visionary' component for smaller scale undertakings in a broad variety of fields. The potential exists of vastly superior performance, since the system can use information of the inference to guide appropriate forms of data access and exert control (in terms of feeding information into a query optimiser) that is not possible given existing architectures. Most work in this field is of the 'paper model' type seeking to generate possible research avenues to assist in the longer-term objectives. The implications for TDS are therefore limited, but may be of bearing to TMD in the > 6 year timescale.

5.2 Activities and Resources

5.2.1 Coupling

UK Activity

Logica, Coopers & Lybrand - Loose coupling : experience gained through the implementation of KBS systems with a DB component in commercial applications. Performance limitations, tuning strategies.

Other Activity

Luc Steels, Brussels University - Esprit work on Intelligent workstation, attempt to treat knowledge as data.

Tight-Coupling. Product Vendor operational research into buffering strategies and techniques for mapping internal KBS representations on to databases eg. AICorp, Waltham, Mass 'transduction filtering'.
AION- suppliers of ADS. Awareness of practical performance issues of loosely-coupled, SQL-based applications with RETE-like features.
Intellicorp - KEE-connection.

Logic Programming - Relational DB coupling is referred to under AI-enhanced database ("Deductive databases").

5.2.2 AI-Enhanced Database

UK Activity

Aberdeen University - Peter Gray. Logic programming / relational integration. Use of Prolog for database 'metadata' in distributed systems.

Imperial College - Kowalski, Kramer, Sloman. Logic programming / relational integration. Temporal logics.
Bristol University - Peter Warren. Extension to the relational model.
Bradford University - Torsun. Logic Programming / Relational integration. Temporal Logics for database systems.
Linn Smart Computing HADES (Hardware Accelerator for Database Expert Systems) uses automatic swapping in and out of memory based on the Recursiv architecture.

IED4 1391 Deductive Object Oriented Databases Heriot-Watt University

Other Activity

Postgres - Stonebraker. Extensions to the Relational Technology INGRES product
NASA - Langley Research Centre. Interdisciplinary Research Office. Adding logic programming inference to pre-existing database systems. Rogers 1987.

ESPRIT 1133 ISIDE Advanced Model for the integration of DB and KB management systems.

ESPRIT 2424 KIWIS - Advanced Knowledge-Based Environments for Database Systems.

5.2.3 Intelligent Front-End / Natural Language

UK Activity

Leeds Polytechnic - Longstaff and Deen. User modelling for intelligent database systems
HP-Labs - Kidd et al. User modelling for KBS

Glasgow University - Van Rijsbergen . Natural Language, KBS and database interaction in the context of information retrieval

ESPRIT 892 DAIDA- BP Advanced interactive development of data-intensive applications.

Other Activity

ESPRIT 311 ADKMS - Advanced data and Knowledge Management Systems using a NL interface to large databases.

Availability of Resource

Glasgow University.

Logica (HCI)

5.2.4 Novel Database Structures

UK Activity

OODB. Glasgow University - Atkinson. Relationship of Object-Oriented to other DB architectures
Deductive Systems. Founded 1984 (Generis) McGregor (ex-Strathclyde) OODB with 'active' capability under Unix or VMS. 'English-like' Intelligent Query Language.
Linn Smart Computing (Recursiv) Objects mapped onto persistent store automatically swapped in and out of memory. Harland 1988
ICL - PISA (Persistent Information Space Architecture) attempt to treat objects the same irrespective of persistence. ESPRIT funded.
University College London - SPAN. Integration of symbolic and numeric computation in highly parallel architectures (ESPRIT 1588). OOF (Object Oriented Framework).

Other Activity

MIT - Actor Languages. All objects classified as Actor (can only act on other objects), Agent (can act and be acted upon) and Server (can only be acted upon)
Ontologic - ONTOS (VBASE, OB2). Supplier with awareness of shortcomings of earlier OODB lacking flexible memory management
Symbolics - Statice. LISP-based system. Uses serialisable transactions for concurrency
Servio Logic - GemStone - Client-server VAX-based OODB commercial product since 1987

5.2.5 Dynamic Databases

UK Activity

Software Sciences - DIOMEDES. High performance real-time distributed DB employing optimistic locking
Heriot Watt - Leitch. KBS toolkit for real-time process control
Glasgow University - Howell. Real-time distributed intelligent knowledge-based control
Cambridge Consultants - MUSE applications
Imperial College - Kowalski/Ferranti. Equator project (ESPRIT-funded) looking at real-time / logic programming extensions.

Other Activity

Univ. of Mass. - Lessing. Blackboard architectures for real-time
Heddaway (1989) - DARPA/Office of Naval Research funded work on variable-precision logics

5.2.6 Knowledge Base Management Systems

UK Activity

HP-labs, Bristol. KB-DB system using Logic Programming and meta-language features.

Other Activity

ESPRIT I ISIDE Advanced model for the integration of DB and KB management systems

ESPRIT II 2443 STRECH. Extensible KBMS for large knowledge-Base

GTE Labs - Brodie. KBMS structural integration

6 DEVELOPMENT METHODS

6.1 State of the Art

KBS Specification

Formal KBS Specification

Very little work has addressed the problem of specifying KBS, though much effort has been put into the provision of specification methods for conventional computer systems by industrial software developers as well as academics. The principle behind these methods is to handle the complexity of the system specification, and in some cases the implementation process, by decomposing them into smaller problem units which can be individually handled. These methods fall into two camps: pragmatic (or structured); and formal.

Pragmatic methods provide a well-structured framework for eliciting and recording system requirements, and in some cases for system implementation. The notation used for specifying the system is semi-formal comprising various types of diagrams and structured documentation. 'Cookbook' like guidelines are provided for specifying the required system and in some cases implementing it. Pragmatic methods have been widely used with varying degrees of success over a number of years. Typical examples are CORE requirements analysis method, IDEF (SADT) and RAPPORT (RAPIDE, RAPIER, RaSQL). For some of the methods support tools have been developed, such as CORE Analyst (SDL, UK) and CORE Workstation (BAe).

With formal methods for specification of the required system, mathematical techniques are adopted to build models of the required system behaviour. Given the formality of the notation specifying the system it is possible to use mathematical techniques, such as theorem proving, to check the specification. One of the major objectives in formal methods is to transform the specification into software automatically; however, this is still a major research issue. Typical examples are Z, VDM (Vienna Development Method) and CSP (Communicating Sequential Processes). Again, tools are under development or developed to support these methods, such as the VDM Toolset (STC and Alvey) and ISAR VDM Workbench (IST).

Formal methods revolve around imposing a fixed structure on system specification and implementation. This is suitable if the required system's functionality is understood and bounded. However, KBS technology is being applied to problems which are complex and ill-defined, requiring the solution to be flexible. As a result, any specification of a KBS must minimise the imposition of an over constraining structure. One approach gaining popularity is the object-oriented approach to the design of KBS. This involves mapping abstract or concrete entities and their associated properties and relations to one another onto an arrangements of 'objects'. These objects form the basis of the specification notation and can easily be implemented using an object-oriented programming language.

Verification and Validation

The objective of validation is to confirm "we have built the right system", that is, the system "fits the purpose", and deals with the properties of the system. For example, for a specification, we would ask "have we specified the right system?", and for an implementation, we would ask "have we implemented the right system?".

The objective of verification is to confirm "we have built the system right", and deals with the properties of the system's development. For example, for a specification, we would ask "is the system specified correctly?", and for an implementation, we would ask "is the system implemented correctly?".

In conventional software engineering the main focus of validation is to ensure the specification adequately describes a suitable system and that of verification is to systematically make certain that the implementation, ie. the software, realises the functional specification. A similar view is taken with verification and validation of KBS. However, the problem with KBS is that the required functionality is extremely difficult to specify and they are extremely difficult to comprehensively test (consider the analogy of assessing a human expert). Thus, though the basic principles of verification and validation for KBS are the same, how it is actually carried out is open to question.

To date little work has been conducted on the validation and verification aspects of KBS. Major activities to be considered worthy of note are:

- The research carried out by Logica for RAE. This has consisted of a study into the validation and verification of real-time KBS and subsequently an implementation prototype VORTEX, in which the recommendations of the study have been applied to the development of a real-time KBS application in the avionics domain;

- The ESPRIT-II VALID project led by Cognitech. This is concerned with the a posteriori validation of KBS and is aiming to develop a set of tools for validation of KBS which can be readily interfaced to applications implemented using a range of KBS tools.

KBS Development

KBS Life Cycle Model

A large amount of work has been invested in developing a methodology for KBS development in order to enable reliable and suitable KBS to be developed, and to allow KBS development projects to be more effectively controlled. The main objective of this research is to move KBS development from a black art employing ad hoc techniques into an engineering discipline with good working practices.

Work has mainly concentrated on outlining a KBS life cycle covering the phases of problem analysis, prototyping, specification, knowledge acquisition, system build, testing, delivery, maintenance, and finally, system disposal. Researchers have tried to identify the boundaries between phases; the sequence of phases; and, for each phase, its inputs and outputs, documentation and the role of participants (knowledge engineer, user, expert, and sponsor). The phases of knowledge acquisition and rapid prototyping have been highlighted as key to any KBS development. A great deal of work has concentrated on these areas. Some researchers have developed computer-based tools to support their life cycles.

Notable work in this area is the KADS Esprit project resulting in the development of a methodology for development of knowledge-based systems and tools to support the methodology. Also, the GEMINI project has carried out significant work in developing a KBS methodology integrated with SSADM (Montgomery 1988).

Knowledge Acquisition

The amount of research work being carried out in knowledge acquisition has exploded in the last few years. Even though AI research has resulted in a number of powerful programming techniques for implementing KBS, such as rule-based and object-oriented programming, the problems of knowledge acquisition, that of abstracting and encoding knowledge within the system, has not been overcome. This has commonly been referred to as the "bottleneck" in KBS development. However, research in this area is beginning to bear fruitful results. To review the area to any reasonable depth would result in a massive tome, therefore a brief overview of the field is given. Further details can be found in the survey papers (Boose 1989) and (Harmon). The main topics emerging are:

Manual Methods

This is a group of methods where a variety of interviewing and psychological techniques are employed by a knowledge engineer to elicit and structure knowledge from one or more experts. They fall into the following categories :

- Interviewing: Various types of approaches to interviewing an expert are used as a basis for eliciting knowledge. Two commonly quoted techniques are the unstructured and structured interviewing. In structured interviewing, general questions are asked with the hope of obtaining suitable answers, everything is recorded and analysed afterwards. Whereas, in the structured approach, a list of questions is strictly adhered to.

- Active knowledge engineer roles: The knowledge engineer plays an active role rather than passively interviewing. In the participant observation, teachback interview and tutorial interview methods the knowledge engineer takes on the roles of apprentice, paraphraser and tutee respectively.

- Brainstorming: These methods rely on the expert rapidly generating a large number of ideas which are recorded and analysed afterwards.

- Psychology based methods: The majority of methods have their roots in the field of psychology. Commonly quoted methods are card sorting, repertory grid and protocol analysis. In card sorting objects on cards are sorted into various categories by the expert to structure the knowledge. The repertory grid method where the expert groups related objects, situations,events according to his or her own criteria. In protocol analysis the knowledge engineer observes and records the expert actually doing the job.

Computer-based Tools

Various computer-based tools have been developed to support the knowledge acquisition process. They tend to fall into the following categories:

- Decision tree systems: These are tools for aiding the creation of fault or decision tree based systems.

- Knowledge structuring tools: Initially in the knowledge acquisition process, the knowledge engineer is presented with a jumble of unorganised data. Tools have evolved to organise and structure this information so it can be incorporated into a system.

- Repertory grid systems: As indicated by their name, these tools have been developed to support the repertory grid manual method.

- Rule induction techniques: Much interest has recently sprung up in combining the application machine learning techniques to the knowledge elicitation task with the objective of automating knowledge acquisition and ultimately removing the need for the knowledge engineer. The most widely reported approach is rule induction where an induction algorithm is applied to a set of examples to generate decision trees or production rules.

- Problem- or Domain-specific Tools: One approach to simplifying knowledge acquisition is to develop a tool specialised for a specific type of KBS. The specialisation may be by incorporating the problem-solving paradigm or domain concepts within the tool so the tool can guide the process, and structure and check elicited knowledge. An example is the tool SALT described earlier.

Intermediate representations

In most cases the knowledge elicited from an expert is in a non-executable form. A large gulf exists between the representation of the raw elicited knowledge and that which is executable by a computer. The problem is to breach this gulf, that is, translate the knowledge from one representation to the other. An approach that has very recently attracted a great deal of interest is the use of an intermediate representation mediating between the expert's and computer's representations. Examples of such representations are diagrams of various forms (eg. circuit diagrams), systemic grammar networks, and semantic nets (Young 1987).

KBS Development Tools

Very few tools have been developed specifically for KBS verification and validation. However, many of the tools which the developer had originally intended for knowledge acquisition perform tasks associated with verification and validation, and should be considered. There is a large overlap between the two areas. The tools tend to fall into two categories: generic and application-specific.

Tools have been developed for various knowledge engineering environments to check knowledge bases. Generally the tools are only applicable for a specific environment, but as many of the environments adopt common knowledge representation formalisms, typically rule and object-oriented representations, the checks performed by the tools are very similar, and the tools can be viewed as generic. Typical checks are:

- Syntax and grammar checkers.
- Rule base consistency to identify conflicting rules, redundant rules and rule subsumption.
- Rule base completeness.

The second tool category is application-specific tools of which two types can be distinguished: tools incorporating the problem-solving paradigm; and high-level representation tools. The first type can mainly be found in the knowledge acquisition field and is based on the tool 'knowing' how the knowledge elicited from the expert is going to be used in the problem-solving process. With such 'awareness' it can flag potential problems in the knowledge base, such as inconsistencies and incompleteness, during encoding of knowledge. An example of such a tool is SALT, a knowledge acquisition tool for expert systems using the *propose-and-refine* problem-solving paradigm. SALT aids in eliciting and checking constraints from an expert by identifying constraint loops, antagonistic constraints and the points where fixes (resolution heuristics for constraint conflicts) are needed. High-level representation tools rely on raising the knowledge representation formalism for domain and inference knowledge to such a level that the expert can directly enter knowledge into the system and check it.

Robust Architectures

In developing safety critical systems, system architectures have been developed with the aim of ensuring continuous reliable operation, ie. tolerance to faults, and graceful degradation of functionality in the presence of faults. A variety of mechanisms to achieve these objectives have been introduced, for example, standby or backup systems, voting mechanisms and watchdog timers.

To-date most of this work has concentrated on ensuring reliability and robustness at the hardware and programming language levels via the deployment of mechanisms implemented at these levels. With the advent of AI, a third level has been added to the technology hierarchy. Interest has recently emerged in using AI techniques to develop new mechanisms to provide tolerance to faults present in the hardware and programming language levels, and in making sure that this additional technology level itself exhibits robust and reliable behaviour.

An example of the use of AI techniques to overcome hardware faults is PDS, a rule-based architecture for diagnosing malfunctions in machine processes from sensor-based data. PDS can continue reasoning in the presence of spurious readings and sensor degradation (Fox 1983).

Approaches are being experimented with, particularly in the Distributed AI sub-field, to get AI systems to monitor their own problem-solving performance and adaptively change their problem-solving approach should the the current one be inefficient. One system exhibiting this behaviour has been implemented on the Distributed Vehicle Monitoring Testbed (DVMT) which employs a meta-level control component to set control parameters guiding the reasoning process (Hudlicka and Lesser 1984). The meta-level control component resets these parameters if certain reasoning performance criteria are not met, for example, if the system was not performing highly rated sub-tasks quickly enough.

KBS Maintenance

Experience with large first generation KBS, such as XCON (formerly R1), shows the issues of knowledge base maintenance are significant and can be a deciding factor in maintaining an operational system. Only with knowledge base maintenance will KBS be operationally deployed. In addition to keeping a KBS operational, there are extra benefits in having easily maintainable KBS:

- Modifications due to new system requirements can be easily incorporated reducing redevelopment effort and costs.
- Maintainable KBS are more easily understood leading to reduced effort in knowledge acquisition and increased user confidence.
- The system can be more extensively tested resulting in more reliable and predictable performance.

Very little work appears to have been done in the area of KBS maintenance. However, it is widely considered that the issue of system maintenance for KBS should be addressed throughout the complete life cycle of the system and not something that follows on from initial development. Viewing maintenance in this context, there is considerable related work, notably:

- Life cycle/Methodology: Much interest has emerged in developing a systematic methodology for constructing KBS. As described previously, methodologies developed to-date are based upon some form of KBS life cycle model which comprises of a number of phases from problem analysis through to system disposal. Associated with each of these phases is a documentation set describing the results of the phase and their relation to previous and subsequent phases. This provides a necessary basis for system maintenance after its deployment.

- Intermediate representations: One problem associated with the maintenance of KBS is how to document the knowledge held within the system in a form comprehensible to both an expert and a knowledge engineer so that knowledge base updates or knowledge bug fixes can be made efficiently. To date most systems rely on documentation comprising a mixture of arbitrary text and machine executable code. One approach to improving this situation is to document the knowledge in an intermediate representation, that is, a representation mediating between the expert's and the computer's representations. This topic is addressed in further detail under knowledge acquisition.

- Tools: A number of tools have been developed for checking knowledge base consistency, as described in the earlier verification and validation section. These would clearly be of great assistance to a knowledge engineer maintaining a knowledge base.

Machine Learning

Machine learning is one of the more active sub-fields of AI with considerable activity on-going within both the US and European scientific communities. Work on rule induction can by now be considered as a mature technology within AI, there being a number of products based upon it and deployed application systems developed using it. There are many systems based upon the archetypal rule induction algorithms namely ID3 and AQ.

Notable developments of these in the European arena are:

- The ESPRIT-I project INSTIL, carried out by GEC-Marconi, University of Paris and Cognitech. This project integrated variants of ID3, AQ11 and a generalisation-based learning system called MAGGY, the integration being both at the algorithm level and via a common representation language called GOL.

- ESPRIT-II project Machine Learning Toolbox (MLT) project. MLT is a major European research programme involving a consortium of 10 partners lead by Nixdorf. The objective of MLT is similar to that of INSTIL but on a grander scale, setting out to provide an integrating framework at the representational level which can be used to bring together the results of machine learning research throughout Europe.

Other more advanced approaches to machine learning such as case-based learning (CBL) and explanation-based learning (EBL) are beginning to mature within the research laboratories and can be expected to be seen in application systems in the next few years. Notable work in this area within Europe has been carried out by Yves Kodratoff's group at the University of Paris in their DISCIPLE system.

6.2 Activities and Resources

6.2.1 KBS Specification

UK Activity

SD-Scicon: CORE specification method for conventional real-time systems.
British Aerospace: CORE workstation.
Praxis: Use of VDM and Z, formal methods in general, safety critical systems.
Logica: Use of VDM and Z, object-oriented specification methods, specification in general.
Han Reichgelt, University of Nottingham: Alvey Socrates project, use of logic as a knowledge representation.
ARE: Study into use of formal methods for KBS.

Other Activity

GMD: ESPRIT-I ToolUse project - specification of conventional software.
CERT: ESPRIT-I ToolUse project - specification of conventional software

Availability of Resources

SD-Scicon, University of Nottingham, Logica, Praxis, British Aerospace.

6.2.2 Validation and Verification

UK Activity

Logica: RAE validation and verification study, VORTEX.
Turing Institute: ESA KBS validation study.
British Aerospace: ESA KBS validation study.

Other Activity

Cognitech: ESPRIT-II VALID project - a posteriori KBS validation
CRI: ESPRIT-II VALID project.

Availability of Resource

Turing Institute, Logica, British Aerospace

6.2.3 KBS Life-Cycle Model

UK Activity

SD-Scicon: ESPRIT-I project 1098, KADS methodology user, CCTA Gemini methodology project.
STC: ESPRIT-I project 1098, KADS methodology user.
Logica: CCTA Gemini methodology project.

Other Activity

B. Wielinga, University of Amsterdam: KADS methodology.
I. Solvberg, Sintef: METAKREK KBS methodology.
GMD: KRITON.
Lockheed: Methodology for real-time KBS?

Availability of Resource

SD-Scicon, Logica, STC, University of Nottingham.

6.2.4 Knowledge Acquisition

UK Activity

Nigel Shadbolt, University of Nottingham: Alvey project on evaluation of knowledge acquisition techniques, ESPRIT-II ACKnowledge project, IED project with HP Labs on PC-based knowledge acquisition tools.
GEC-Marconi Research Centre: TACKTIX repertory grid system, ESPRIT-II ACKnowledge project - integrated knowledge acquisition workbench.
Open University: Keats knowledge acquisition tool.

Other Activity

Boose and Bradshaw, Boeing: ETS, Aquinas, use of repertory grids for knowledge acquisition.
Gaines and Shaw, University of Calgary: KSS0, Kitten, use of repertory grids for knowledge acquisition.
Mark Musen, Stanford University: OPAL, Protege.
Neuron Data Inc: Nextra repertory grid based front-end to Nexpert expert system tool.
University of Amsterdam: KADS Power Tools, Shelley, ESPRIT-II ACKnowledge project.
Sintef: METATOOL knowledge acquisition tool, ESPRIT-II ACKnowledge.
GMD: KRITON.
Katarina Morik, Technical University of Berlin: BLIP.

Availability of Resource

University of Nottingham, Open University, GEC-Marconi Research Centre.

6.2.5 KBS Development Tools

UK Activity

Logica: VORTEX.

Other Activity

John McDermott, Carnegie-Mellon University: MORE, MOLE.
Mark Musen, Stanford University: OPAL, Protege.

Availability of Resources

None identified.

6.2.6 Robust Architectures for KBS

UK Activity

A number of successful products exist in the area of fault-tolerant systems, eg Stratus/Tandem. In fact most operational, integrated systems can be considered to fall into this category. Whilst these address a number of the overall architectural goals identified in this substream there are major issues concerned with robustness of a KBS's reasoning processes for which no significant activity can be identified.

Other Activity

None identified

Availability of Resources

None identified

6.2.7 KBS Maintenance

UK Activity

Keith Bennett, University of Durham: Centre for Software Maintenance.
ICL: IED MAKE proposal.

Other Activity

None identified

Availability of Resource

ICL, University of Durham.

6.2.8 Machine Learning

UK Activity

Turing Institute: DUCE machine learning system, ESPRIT-II Machine Learning Toolbox project, machine learning in general.
GEC-Marconi Research Centre: ESPRIT-I INSTIL integrated machine learning system.
British Aerospace: ESPRIT-II Machine Learning Toolbox project.
Derek Sleeman, University of Aberdeen: ESPRIT-II Machine Learning Toolbox project.

Other Activity

Tom Mitchell, Carnegie-Mellon University: Version spaces etc.
Yves Kodratoff, University of Paris: ESPRIT-I INSTIL project - integration of symbolic and numeric learning systems, DISCIPLE explanation-based learning system, ESPRIT-II Machine Learning Toolbox project.
Intellisoft (France): KATE integrated machine learning system, ESPRIT-II Machine Learning Toolbox project.
Katarina Morik, Technical University of Berlin: BLIP.

Availability of Resource

Turing Institute, GEC-Marconi Research Centre, British Aerospace, University of Aberdeen.

7 TMD APPLICATIONS

7.1 Overview of the TMD Domain

The TMD scenario initially considered in this study was that for the European theatre which has been developed by the United Kingdom Architecture Study (UKAS) for their programme of work. As a result of discussions with SDIO this was modified to consider the wider implications of the CONUS SDS (Continental US Strategic Defence System) phase 1 architecture. In this description the components of the architectures are discussed as they are considered to be a direct consequence of the scenario.

The UKAS scenario includes the political and defence concepts for a TMD system to be installed in its initial phase in 2005. Thereafter it can be expanded in a series of phases to cover the development pull caused by a reactive enemy offensive technology and a technology push as further weapon and sensor systems, and other technologies become available.

The latest UKAS phase 6 scenario is adopted. This considers the anticipated effect of the Soviet implementing a revised global offensive policy following the signing of the INF treaty.

The scenario is described in the Final Report of the UKAS Phase 6, July 1989, particularly:

Volume 1.2 Overview
Volume 2.3 Track Processing Study
Volume 2.5 Concept of Operation
Volume 4 Architecture Spec
Volume 5.1 System Level Spec
Volume 5.4 BM/C3 Sub-system Spec
Volume 7 Operational Deployment Concept

These documents are classified.

The CONUS SDS Phase 1 system is also designed to defend priority assets against a worst case all out ballistic missile attack. The architecture consists of a number of integrated systems. These are:

For surveillance:

- a satellite based Boost Surveillance and Tracking System (BSTS) and a Space Surveillance and Tracking System (SSTS), both using light sensors over a spread of wavelengths.
- a Ground Based Radar (GBR) system, and a Ground-Based Surveillance and Tracking System (GSTS) consisting of probes.

For interception:

- Ground Based Interceptor (GBI) and Space Based Interceptor (SBI) systems which are hit to kill interceptor vehicles (IV).

These are all autonomous systems controlled via Command and Control Elements (CCE) which is a network of facilities that ensure that 'human-in-control' exists during all SDS operational states. The structure of CCE is centralised command and decentralised execution. Each contributing system is autonomous but cooperating with Course of Action (COA) revisions from the CCE.

The scenario and system for CONUS SDS Phase 1 is described in the secret document Strategic Defence System Phase 1 System Description (UK Release), F144-23-2, December 1989.

b) Comparison with naval scenario

A comparison of the TMD and naval scenarios is best performed through a comparison of the attributes related to the entities which are processed by the functions and sub-functions in the two systems. These functions as identified from the literature for the two systems are:-

TMD	Naval
sensor hand-on	
track processing	picture compilation
trajectory / impact point prediction	track prediction
association	association
correlation	correlation
discrimination	(track identification)
kill assessment	
threat assessment / raid analysis	threat assessment
situation assessment	situation assessment
adaptive preferential defence	reactive resource allocation
weapon (to Target) assignment	weapon assignment
adaptive firing doctrine	
planning	planning
comms network management	comms network management

and the supporting issues of:

layered defence	layered defence
processor throughput	
distributed system	own ship is basis of system
	Human Computer Interface (HCI)
new architectural concept	evolutionary architecture
single domain	multi-domain

The two lists have been produced by two disparate groups. The TMD developers had a clean sheet of paper and considered how defence could be achieved given adequate technology. Their brief was to drive (push) the technology. The naval C^2 system developers were in an evolutionary environment attempting to perceive how emerging technology (pull) could support the required functions. There was no effort directed towards merging the two sets of functions, in that functions with the same name might only be superficially related, while seemingly dissimilar functions are attempting to achieve the same end result in a similar situation.

The most obvious difference between the two scenarios is the characteristics of the threat. The TMD threat is ballistic and therefore the object tracks are highly predictable. The threat's opportunity for evasive manoeuvre is small. Where it can be achieved the manoeuvre is limited. Each RV will be accompanied by 'many' decoys, which will be ballistic, but most will 'strip-out' at the early stages of re-entry. The overwhelming number of decoys in the mid-course will drive processor throughput. The ballistic calculation is not a candidate for the application of AI technology although the control of de-clustering and cluster fusion between sensors is a candidate. In the naval AAW warfare domain less than 20% of a threat are 'intelligent' manned aircraft, the rest are missiles with limited intelligent manoeuvre ability. Although the two forms of track are different, both have an element of pre-programmed intelligent manoeuvre, which may be amenable to prediction by AI techniques. AI techniques can be used here to apply the information obtained on the manoeuvre strategy of the vehicle, once its type has been discriminated.

Track prediction is required for sensor hand-on or the cueing of sensors. Association of plots within a sensor to form a track, and the correlation of tracks between sensors is assumed to be 'possible' within the (UKAS) TMD scenario. Considerable work is being performed in all countries participating in SDI on all contributing technologies (hardware architecture, chip design, algorithms, etc) to achieve this possibility.

The most common method of numeric solution for track prediction is through fourth order Kalman filters or third order polynomials (for the terminal phase). AI techniques are not favoured because the problem can be easily expressed in mathematics. Some of the sensors are designed to measure attributes of the object (eg. radar cross section, radar scintillation, IR temperature, IR scintillation, etc) and this data is also sent to the processing nodes to aid in the discrimination process. This is a mathematical application of uncertainty hypothesis equations (typically Bayesian logic with extensions by Shafer and Dempster) to discriminate the type of object (ie. RV, decoy, PBV, junk, etc).

Within the naval environment the process of association and correlation relates to radar plots/tracks and ESM bearing (including attributes such as IFF/NIS response) from sensors within a ship and to correlate these with track positional data from cooperating units. This data is 'fused' further with plan and encyclopedic data to identify the nature of the object. In current practise the whole process, called picture compilation is most unsatisfactory and requires an intelligent, highly trained and skilled operator to solve. Picture compilation is thus seen as a process highly amenable to AI which has led to the work on the TDS.

Kill assessment has an important role in the TMD scenario since an object which is in an exo-atmospheric ballistic trajectory does not explode when hit, unless the kinetic energy of the impacting missile is greater than some critical value. Instead small changes of state have to be measured and combined with own engagement records (in real time) using a mathematical process similar to that for discrimination. There is no immediate parallel in the naval scenario except for the disappearance of a plot/track/bearing. This brings the concept in line with naval procedures.

Threat assessment (which is the assessment of the combined effects of each threat object with the inferred enemy intentions), and situation assessment (which includes knowledge on status of own forces to organise a response) are somewhat similar in nature in both scenarios and both are amenable to AI techniques. Raid Analysis is another term used for these processes.

Reactive resource allocation (in naval terms) and Adaptive Preferential Defence (in TMD terms) both relate to the defence objectives of the force to the current situation. As naval High Value Units (HVUs) or TMD priority defended items are lost, or a raid is seen to be concentrating on a type within the priority/value list, then weapon assignment rules or Course of Action (COA) are changed to optimise the defence against the threat so as to best achieve the overall mission objectives given the current situation.

Planning is common to both scenarios and is the function most obviously amenable to AI techniques. Planning includes pre-hostilities, strategic, tactical (long, medium & short time frames), logistic, BMD engagement during the battle as well as Reconstruction and Recovery (R2) after the battle.

There is considerable scope in the naval scenario for tactical planning because of the different naval warfare domains. These are air, surface, sub-surface and even EW warfare environments. In addition the movement and tasks within a manoeuvre force over a period of many days can be planned. However pre-engagement planning within the (UKAS) TMD is more to do with dispersal and coverage to support the mobile ground forces than in the exchange during a number of alternate set piece nuclear exchanges.

Supporting EW will be present in both scenarios, although more effective in the naval scenario because the jammers will be closer to own radars, and the incoming missiles in the naval case will contain a better counter-measure suite.

Both scenarios have communication network management problems which are a extremely complex in a nuclear and electronic warfare environment. Yet the performance of a distributed system relies on it. AI techniques are currently being investigated for real time optimisation of comms networks in current systems.

As regards Human Computer Interface (HCI), this is an integral part of the naval C^2 operation and the man is very much 'in-the-loop' of each timeframe, for the picture compilation and weapon response loop, as well as the slower manoeuvre response.

7.2 Activity and Potential Sources for Research

UK Activity

All SDI work in the UK is managed or sponsored by the SDIPO, who either directly control the industry consortium or delegate some project management functions to the research establishments.

The SDI activity in the UK initially stemmed from the United Kingdom Architecture Study (UKAS) which, up to the end of its phase 4, had contributions from 22 of the UK defence companies and from MOD research establishments. The key players during these phases were:

Research establishments:

RAE - Special Systems Group, Farnborough

RSRE - Air Defence Group, Malvern

Industry contractors with area of significant contribution in brackets:

British Aerospace, Stevenage (weapon sub-system design)

Marconi Radar, Chelmsford (radar sub-system design)

Plessey Defence Systems, Christchurch (BM/C3 sub-system design, algorithms, Concept of Operations)

Hunting Engineering Ltd, (HEL), Ampthill (systems architecture design, develop and maintenance of the TLSM, development of the threat characteristics)

Thorn EMI (IR sensor sub-system design)

Theta Analysis and Systems Ltd, Aldershot (cost analysis)

Logica SDS Ltd, Cobham (integrated activities with Plessey and HEL, design direction of BM/C3 systems architecture)

Software Science Ltd, (SSL), Farnborough (sensor sub-system design, track processing algorithms)

Admiral Computing Ltd, Camberley (support to PA)

PA Consultants Ltd, London (project management, concepts)

In phases 5 & 6 of the study the scope was reduced to the SDIPO directly managing Plessey, BAe, HEL, PA, Theta and SSL. It is the output of phase 6 of this study which is the baseline architecture for this research.

A contributing and parallel study to UKAS phases 1 to 4 was the European BM/C3 study by RSRE with Easams, Ferranti and System Designers. A particularly useful output of this study is the functional analysis of a generic theatre defence BM/C3 architecture.

A further area of work in the SDI envelope of studies is the work on the UK node of the European Air Defence Test Bed (EADTB). This work is being undertaken by RSRE who are leading SSL, MEL and HEL. This forms part of an overall plan to link test nodes in each of the participating European countries to some in the USA, and then send to the US SDI National Test Bed (NTB).

As a spin-off from the UKAS study RAE are leading SSL in a study into the use of AI techniques in a discrimination framework. It is understood that this study is concerned specifically with the demonstration of blackboard architecture to utilise measured threat object attributes to discriminate RVs.

7.3 Key Technical Issues

Summarising from the above discussion the key technical issues with respect to TMD BM/C3 are (in approximate priority order):

- Object discrimination including kill assessment
- Track correlation including:

  frame to frame processing

  sensor to sensor processing

  element to element processing

- De-clustering of threat objects and the cluster fusion between sensors. This includes the maintenance of clusters relating to RV threat group.
- Maintenance of the communications network in a hostile environment and timeliness of data
- Distributed data fusion processing

Although all the technical issues (even minor ones) in TMD are currently beyond the state of the art of current systems, these issues are crucial to the feasibility of the system. Not all these issues are completely amenable to AI techniques.

7.4 Potential for AI Technology

The various applications that are potential candidates for AI techniques, which have been identified in the above discussion are given below:

- Planning (including pre-hostilities, strategic, tactical (long, medium & short time frames), logistic, BMD engagement during the battle as well as Reconstruction and Recovery (R2) after the battle)
- De-clustering and cluster fusion between sensors
- Track prediction of intelligently manoeuvring objects
- Discrimination (already subject of RAE/SSL SDIO contract)
- Kill assessment
- Raid analysis including threat assessment and situation assessment
- Sensor coverage adaptive strategy
- Weapon coverage adaptive strategy
- Adaptive modifications to the Course of Action, Weapon Target Assignment strategy, Preferential Defence strategy and Firing Doctrine
- Human Computer Interface to assist in battle management by man-in-control
- Communications network management

There are three levels for potential AI technology. Those related to:

a) The limitations of the TMD demonstrator

b) UKAS Design

c) TMD in general (ie UKAS plus CONUS SDS)

The table (below) gives the potential for AI solutions within the constraints of these levels.

Level	TMD demo	UKAS	General
Applications:			
Planning			
pre-hostilities / strategic	no	*	yes
tactical / engagement	no	yes	yes
logistic	no	*	yes
recovery & reconstitution	no	*	yes
De-clustering	no	yes	yes
Track prediction of manoeuvring object	yes	no	yes
Discrimination	no	yes	yes
Kill assessment	yes	yes	yes
Raid analysis	yes	yes	yes
Sensor coverage	yes	yes	yes
Weapon coverage	yes	yes	yes
Man-in-control:	limited	yes	yes
Course of Action			
Weapon to Target Assignment			
Adaptive Preferential Defence			
Adaptive Firing Doctrine			
Human computer interface	yes	limited	yes
Communications	no	yes	yes

* indicates that UKAS requirement is not defined.

Of those applications which are suitable for the TMD demonstrator level, the following AI research topics are relevant:

Track prediction of manoeuvring object:

- real time system design
- spatial knowledge manipulation
- adversary knowledge manipulation
- uncertainty knowledge manipulation
- machine learning

Kill assessment for hard kills:

- distributed AI
- real time systems design
- uncertainty knowledge manipulation

- knowledge base management systems
- machine learning

Sensor and Weapon coverage:

- spatial knowledge manipulation
- adversary knowledge manipulation
- uncertainty knowledge manipulation
- planning knowledge manipulation

Man-in-control:

- spatial knowledge manipulation
- adversary knowledge manipulation
- uncertainty knowledge manipulation
- planning knowledge manipulation
- distributed AI
- real time systems design

Human Computer Interface:

all HCI aspects.

In addition the following research topics are generally applicable (in a supporting role) to these applications:

- reason maintenance
- dynamic databases
- paradigms
- robust architecture
- KBS maintenance
- KBS validation and verification

8 NAVAL APPLICATIONS

The objective of this research stream is to highlight those application areas that it is felt the Technical Streams should focus upon. The study is primarily interested in those aspects of Command and Control that have similar functional requirements across the two domains and are areas in which research is expected to realise a cost effective return. In very general terms, the underlying problem facing the Command is that of using diverse sources of information to construct a picture of the situation, to evaluate the threat that it poses, and to plan and allocate ones own resources in such a way as to minimise the cost to own forces in accordance with the overall tactical strategic plan.

To ascertain the functional requirements common to both domains it is necessary to understand current tactical naval doctrine and how future sensor and weapon systems may influence this.

8.1 Overview of the Naval Domain

A number of primary and secondary roles can be distinguished for Naval vessels. Of the primary ones, the most significant are :

- providing support for hunter-killer units
- defending convoys
- supporting amphibious operations.

Secondary roles include surveillance, intelligence gathering, disaster-relieve and showing the flag. For the purposes of this study, emphasis has been placed on the primary roles, with particular interest on those areas that are felt to demonstrate most overlap with the TMD domain.

8.2 Activity and Potential Sources for Research

The naval environment is a complex one. It can consist of surface, sub-surface, air and space-borne assets on both sides, their operational effectiveness being heavily dependent on the information sources available.

With the increased performance of sensor systems the amount of data that needs to be processed to form a coherent tactical picture will seriously overload the already hard pressed human element of current generation naval command and control systems.

In addition to the problems created by volume of data, there is a need to realise improvements in performance in the combining of data from disparate sources such as Radar, Optics, ESM, IFF, Sonar and Intelligence reports.

The shortcomings in the performance of naval command and control systems have been recognised by ARE (AXT) and has led to the current research programme investigating the potential of Artificial Intelligence techniques to provide automated support to naval command and control in the form of assistance with the real-time tasks of sensor data interpretation, picture compilation, situation assessment and resource allocation.

The first objective has been to provide automated support to the human operator in forming a coherent tactical picture. This has led to the current data fusion demonstrator programme. The next stage is to build on the tactical picture produced by data fusion and interpret the picture to form a consistent view of the tactical position in terms of the effectiveness of resources deployed under the prevailing conditions.

The objective of the current SDI/ARE research programme is to independently analyse the techniques that could best be applied to advantage to improve the performance and operability of naval command and control. ARE would be a prime source of advice and the data fusion demonstrator a possible vehicle on which to exercise some of the research topics.

8.3 Key Technical Issues

8.3.1 Areas of Applicability of AI

As the key purpose of the study is to determine functional areas of similarity between naval and SDI elements of command and control it is appropriate even at this early stage to focus our attention on Anti-Air warfare extrapolating the threat as it exists today, to that anticipated in the mid 1990's. This provides an approach which will enable us to focus our attention on a manageable subset of naval command and control in which the risk of the ships defences becoming saturated exists, and split second decisions must be made on the best possible information available.

Many of the techniques appropriate to Anti-Air warfare will have applications to Surface and Sub-surface warfare, again with demanding performance requirements but typically with a more relaxed real time response requirement.

8.3.1.1 Anti-Air Warfare

The air threat facing a navy of today may consist of subsonic and supersonic aircraft carrying bombs, air, surface and submarine launched subsonic sea skimming missiles, and supersonic high diving surface launched missiles.

The threat recognised as being of most concern to a blue water navy is that posed by the modern missile, both the sea-skimmers and high divers. Future missile designs are expected to follow the obvious trend of reduced susceptibility to counter measures, increased speed of the missile to reduce the reaction time available to defensive systems and a move towards saturation attacking where a single missile may release guided submunitions just beyond the effective range of the ships defences.

Within the time scales of the anticipated operational implementation of a system designed around the techniques decided upon during the research programme, the air threat with which the navy will have to contend will have much in common with the anticipated SDI threat.

Considering the very simplistic approach used in current systems in evaluating the threat and subsequent assignment of weapon systems there appears to be considerable potential for improving the effectiveness of Naval air defence systems. The introduction of new improved sensors will increase the number of objects detected necessitating the use of improved tracking algorithms to provide improved track discrimination and flight profile information. Sensors will also provide more detailed information on characteristics of detected objects such as estimates of size, propulsion unit characteristics (modulation of the radar pulse), IR, etc. which can be used to help in object identification.

Although it is tempting to consider a truly distributed sensor system capable of relaying real-time track or sensor level data via ship to ship high speed microwave links for subsequent fusion to provide faster track up-date rates and alternate views of the volume under surveillance, there are difficulties. Practical problems exist with respect to biases corresponding to the uncertainty in sensor platform positions, the necessity for significantly higher speed data links to carry the additional track data and associated variance information, and the very large increase in processing power and data throughput required.

It is far more practical to consider ways in which new techniques can be used to advantage in improving AAW performance in line with the Naval policy of support defence where missile ships will be required to provide a more effective defence for ships within their sector of responsibility.

A prerequisite to the optimum allocation of defensive resources is the tactical picture and its correct interpretation. In building up the tactical picture all relevant information, be it sensor data, intelligence data or encyclopedic data, needs to be made use off taking due account of the accuracy of the information if it is known.

This then is the task for the data fusion function. Intelligence information can provide prior expectations of the presence or absence of platform types. The fusion of such data with real-time sensor data can be a complex task particularly if the data is stale in which case knowledge of track history needs to be incorporated together with what is known about operating patterns.

There are a number of ways in which identity data can be combined with varying degrees of success. Techniques include Bayesian, Weighted Voting, Fuzzy Logic, Hypotheses testing, Schafer-Dempster, and others.

The Bayesian approach is perhaps the best established. It is based on probabilistic reasoning which has a sound mathematical basis, but it does suffer from problems such as its inability to reflect in the posterior probabilities any uncertainty associated with poor priors. The same problem applies to uncertainty associated with likelihoods which is not reflected in the computed posterior probabilities. Difficulties also arise in coping with objects that have not been defined as belonging to a prescribed set.

Shafer-Dempster may go some way to providing a possible solution to these problems with its concept that evidence relating to a proposition can be apportioned to evidence in support for, evidence against, and the degree of uncertainty.

8.3.1.2 Anti-Air Ship Defence

The sequence of actions followed by an Anti-Air warfare team when faced by an air threat will depend on the class of ship, sensor and weapon fits and even personal preferences of the team itself. It is therefore only possible to generalise on the sequence of actions which would form the skeleton plan for the defence of a warship and the ships for which she is responsible.

On the detection of a contact, the first task of the combat system is to establish whether it is a hostile and whether it poses a threat to ships in own ships sector of responsibility. IFF may help, as may flight profile, intelligence information and supporting sensor data from own ship and supporting units. If it is a threat, the combat system needs to know its magnitude ie.

- single target or a group of targets
- aircraft or a missile
- type of aircraft or missile

Flight profile data, IR data, intelligent reports, and sonar contacts may all be used to assist in classification of the target.

8.3.1.3 Key Functional Areas

Multi-Sensor Data Fusion

To build up the tactical picture it is first necessary to detect, locate, track and if possible, classify all objects which may conceivably contribute to the tactical situation. Information sources which might contribute to building up the tactical picture include radar, ESM, IFF, and optical sensor data, intelligence data provided by human observers, and encyclopaedic information and operational plans. For the purposes of this report, the term Multi-Sensor Data Fusion refers to the combination of such disparate information sources to provide track and identity information needed for the tactical picture and situation assessment.

Systems to date have adopted simplistic algorithms for the correlation and classification of track data. For both the SDI and future naval tactical environments, new approaches need to be adopted to:

- Reduce the amount of data that needs to be considered and processed
- Improve the accuracy of association, correlation and object discrimination by making optimum use of all available information
- Maintain a single consistent track when the target passes through the coverage regions of different sensors
- Make best possible use of all information when attempting to classify a target.

Some of these requirements appear contradictory e.g. reducing the amount of data but using all the information available. The final design will almost certainly be a compromise, but it is important to appreciate that improvements in signal processing and the application of AI to the back end of the system alone is not sufficient. The whole of the processing chain from sensor data reports through to the fusion of intelligence information needs to be carefully considered.

Tracking

Irrespective of the general approach, the combination of position related information can generally be regarded as a two-stage process. The first stage involves the association or correlation process, the purpose of which is to establish the degree of correspondence between reports from different sensors. Typically, some form of spatial filtering is employed to reduce the number of reports that need to be considered in any specific instance of correlation.

Two fundamental problems that exist from just a single sensor are:

- Two or more reports from the same target due to an overlap of beams
- A single report is possible from multiple targets due to limited sensor resolution.

As targets approach their intended destination, the diverse distribution of multiple sensors and the changing relative position of targets will result in a changing angular separation of individual targets, which in turn will result in a fluctuating number of reports.

This gives rise to three regions:

(1) a region of unambiguous correlation for well separated targets

(2) an unstable region where highly inaccurate tracking can occur

(3) a region for closely spaced targets where miscorrelation occurs, but tracking remains stable insofar as track loss will be infrequent.

Association

We need here to review the process of report to track association. To eliminate unlikely associations a technique called 'gating' is carried out. Problems arise when more than one return is within the track gate or the report is within the gates of more than one track.

The problem can be alleviated by varying the gate size and shape depending on track quality and choosing the most appropriate algorithm for the measurement of separation between reports and tracks.

In dense target or high clutter environments where groups of three or more reports may exhibit a high degree of correlation, improved tracking performance is possible by postponing the decision and forming alternative hypothesis thus allowing future measurements to aid in the correlation decision. This approach of generating hypothesis is often referred to as Multiple Hypothesis Tracking (MHT). It involves generating several hypotheses about which reports actually correspond to the same target, and forming tentative tracks in the process. As time progresses, new hypotheses are made and existing hypotheses are examined and combined, confirmed or rejected.

An alternative approach, the Joint Probabilistic Data Association (JPDA) method is particularly suited to busy environments as it overcomes the problems associated with the storage and processing of the MHT method associated with the rapid growth of the hypotheses tree. This approach also involves the generation of hypotheses about associations of reports to tracks where each hypotheses is assigned a probability. In this case however multiple hypotheses are formed after each sensor scan, and combined before the next scan. The updated estimate for a track may therefore contain contributions from several reports. Whilst reducing the load and storage requirements JPDA suffers from problems of track initiation and track deletion.

In scenarios where large numbers of objects in close proximity need to be tracked, an approach involving the generation of hypotheses is to be favoured. Temporary track loss due to obscuration of objects and sensor coverage problems are better tolerated, and clues as to the identity of individual objects can be associated with the hypotheses to assist in the association process.

In scenarios where the numbers of objects are less, but the clutter can be dense, again, hypotheses based tracking is to be favoured.

It is possible that a hybrid scheme using MHT and JPDA could be used to overcome the initiation and deletion problems associated with JPDA alone.

The concept of group tracking should also be looked at as a means of reducing the processing load to manageable proportions during periods of high activity e.g grouping together objects in close proximity known to be decoys.

Correlation

The output of the data fusion process is a correlated central track database from which data can be made available to the situation assessment process. In a multi-sensor system, sensors will be providing track reports which at some level need to be correlated to provide a single best estimate of the objects being tracked.

One possibility is to have a track table associated with each sensor. Creating a central track table will require the tracks in each of the sensor databases to be correlated to provide one entry for each true object in the coverage of one or more sensors.

The advantages associated with sensor level tracking are:

- reduced loading per processor and higher survival rate due to the distributed architectural approach possible.

- sensor degradation is more easily identified and its observations will not affect tracks of other sensors.

- tracking filters can be tailored to the characteristics of each individual sensor.

There are some disadvantages including:

- less accurate tracking may result compared to the central level tracking scheme due to the less frequent update rates at the sensor track level.

- a Multiple Hypotheses Tracking approach would be more difficult to implement to realise its full potential.

The other possibility is to correlate the reports from each sensor at the central track table to provide a single report for each true object. The reports are then used to update tracks.

The advantages associated with central level tracking are:

- Tracking performance advantages as report data from the various sensors will under different circumstances have a varying ability to confirm or sustain a track.

- The higher data rate will allow more accurate tracking of manoeuvring targets.

- System latency in producing central level tracks will be reduced.

There are also disadvantages with the central level tracking:

- The high data rates and processing load at the central processing level could become a bottleneck.

- Degradation quality of data from a particular sensor could severely compromise the tracking performance of the whole system before the problem becomes apparent.

Classification

The task of classification is to establish the identity of a contact. This is of considerable importance insofar as establishing the contacts hostility. If the contact does represent a threat then its targeting accuracy, range, number of warheads, flight profiles, and warhead lethality are important parameters which may be retrievable from an encyclopaedic database and from which it may be possible to:

- improve tracking accuracy using a more accurate model of target dynamics.

- establish a plan for the optimum deployment of counter measures.

- maintain a record of the enemies remaining assets.

Sensor Hand-On

The most suitable approach to be adopted for sensor hand-on will depend on whether sensor level tracking or central level tracking is used or indeed whether some hybrid scheme is adopted.

In either case, the problem seems to be one of the target leaving areas in which sensor tracking accuracy is diminishing and entering into adjacent areas in which the target can be tracked by other sensors.

The overlap in sensor responsibility for the volume of space to be under surveillance will depend on the relative performances of the sensors at the position in space in question.

As the sensor performance deteriorates to the point that it degrades the accuracy of tracking, the responsibility for tracking the target would be transferred to another sensor.

A balance must be struck between the quality of data being provided on a particular target and the number of targets being tracked by the sensor.

This leads on to the question of sensor allocation i.e. the allocation of sensor resources in a way that maximizes the overall utility. Given multiple potential targets and multiple sensors that may be used to verify or identify these targets, it is possible knowing the conditional probability that the sensor will report a target present given that it is present, and the conditional probability that the sensor will report target present when it is not, to calculate a measure of utility for the sensor-target assignment.

If the utility factors are calculated for all target sensor pairings it is possible to solve for the optimum assignments within specified constraints such as maximum number of targets per sensor.

Situation Assessment

Situation assessment takes the results of the data fusion and attempts to build an intelligent assessment of the environment from a tactical perspective. 'Situation Assessment' is some times taken to subsume 'Threat Evaluation' (which is here considered separately), as well as the following components :

- Assessing engagements to provide outcomes
- Judging threats against Rules of Engagement
- Providing a model of sensor and weapon coverage
- Monitoring how well the current plan is progressing
- Attempting model the extents of the adversaries knowledge

Threat Evaluation

With information provided by the Situation Assessment function, Threat Evaluation is concerned with determining the magnitude of the threat. Without a threat picture it is not possible to allocate resources such as sensors and weapons in an effective manner.

In the SDI scenario, the blue force assets at risk include:

- radar installations
- missile silos
- airfields
- command centres
- submarine bases
- arms depots
- power station

whereas in the Naval environment assets typically consist of:

- tankers
- aircraft carriers
- cargo vessels
- destroyers
- frigates
- submarines

In both environments it is possible to apply the concept of asset value as a building block which may be used in the calculation of the value of the whole force. The following demonstrates the basic principles of using uncertain information relating to type of threat, intended target, effectiveness of the countermeasure etc. in a probabilistic framework to calculate the magnitude of the overall threat. The processing of uncertain information can be carried out in various ways e.g bayesian, Schafer- Dempster, fuzzy logic - these will be discussed briefly at the end of this section.

The following scenario will assist in understanding the basic problem and a viable approach to a solution.

Suppose we have several assets e.g. a task force comprising several warships and two tankers. Each asset has a different value and these values can change with time, i.e. if one of the tankers is destroyed, the value of the remaining tanker increases. The amount by which it increases reflects its importance and contribution in maintaining an operational capability. The value of an asset is dependent therefore on its context. This introduces the concept of encoding information in a form of cost to own force rather than each individual target thereby enabling costs to be associated with combinations of assets destroyed.

In the case of the tankers, the cost of losing one may be tolerable, but the loss of both will completely nullify the ability of the task force to reach its destination.

The problem is therefore to calculate the probabilities associated with possible combinations of assets that will be hit, and combine these with the costs of each outcome to obtain the possible costs of an engagement..

Resource Allocation

Determine which targets are best engaged by SAM, CIWS or GUNS.

SAM has the longest range and highest probability of kill and therefore should be used on all known targets that fall within its missile engagement envelope.

CIWS is effective against aircraft and missiles at short range and has a medium probability of kill. GUNS are most effective against slow aircraft at short range but only have a low probability of kill.

If there is a target against which a jammer may be effective, allocate it.

If a solution has been found in which all targets can be engaged then propose solution, otherwise prioritise targets by:

- targets that fall within own sector of responsibility
- targets aimed at own ship
- other targets within engagement envelope

This strategy is very general, and although reasonably sensible and non-contentious, the defensive strategy is far from optimum. It is perhaps appropriate here to introduce the concept of the value of ships in the context of their role in satisfactorily completing the mission. This provides an approach to prioritising the allocation of defensive weapon systems, such that the estimated 'loss of value' to own force as a consequence of an engagement can be minimised.

The next consideration may be whether to use decoys, and if so, how should they be deployed.

Chaff may have already been deployed and may require re-seeding.

For new targets it has to be established what the target range is and target radar most suitable for:

- single shot distraction shot
- full distraction pattern
- centroid seduction pattern
- seduction dump beyond ship

Consider whether a ship manoeuvre is required:

- to remain inside decoy pattern
- to open arcs of fire
- to present the most difficult target for the threat

If a manoeuvre is required, consider whether it endangers any other ships either by collision or crossing the line of fire of a missile ship.